Technical Report Number 89-10-01

MEMBERSHIP-SET PARAMETER ESTIMATION VIA OPTIMAL BOUNDING ELLIPSOIDS

Ashok K. Rao and Yih-Fang Huang

Department of Electrical and Computer Engineering
University of Notre Dame
Notre Dame, Indiana 46556

OCTOBER 1989

A SEMIANNUAL TECHNICAL
PROGRESS REPORT FOR
OFFICE OF NAVAL RESEARCH
CONTRACT N00014-89-J-1788

MEMBERSHIP-SET PARAMETER ESTIMATION VIA OPTIMAL BOUNDING ELLIPSOIDS*

Abstract

In the last few years, there seems to have been a resurgence of interest in the membership-set-theoretic approach to parameter estimation. This report concentrates on the optimal bounding ellipsoid (OBE) approach to membership-set parameter estimation, with emphasis being placed on the performance of one particular OBE algorithm in non-ideal conditions. It is first shown that OBE algorithms offer distinct advantages over commonly used recursive parameter estimation algorithms like the recursive least-squares (RLS) algorithm in some real-life environments. Then the extension of a particular OBE algorithm to the problem of parameter estimation with unobservable but bounded inputs (ARMA parameter estimation) is discussed in some detail. The problem is important because, in many signal processing applications, the inputs to the system under consideration are unknown. Analysis of the extended algorithm shows that under some conditions, the extended algorithm yields 100% confidence intervals for the parameters at every sampling instant. This feature does not appear to be present in any other existing ARMA parameter estimation algorithms. Furthermore, the transient performance of this algorithm is observed to be superior to that of the extended least-squares algorithm. Finite precision effects of one of the OBE algorithms are also studied via analysis of error propagation in the algorithm and through simulations. The analysis shows that the algorithm is stable with respect to errors due to finite word-length computation and storage. Simulation results demonstrate the superiority of the algorithm to the conventional recursive least-squares algorithm for small word-lengths. Finally, analysis of the tracking characteristics of one of the OBE algorithms is performed. It is shown that the algorithm is capable of tracking small time variations in the parameters. Since large variations may cause the algorithm to fail, a rescue procedure is proposed which can enable the algorithm to also track large time variations. Simulation results demonstrate that the tracking capability of the algorithm is comparable to that of existing adaptive filtering algorithms.

*This report is a reproduction of Ph.D. dissertation of Ashok K. Rao, Department of Electrical and Computer Engineering, University of Notre Dame, Notre Dame, Indiana, August, 1989. This work has been supported, in part, by the National Science Foundation under Grant MIP 87-11174 and , in part, by the Office of Naval Research under Contracts N00014-87-k-0284 and N00014-89-J-1788

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

MEMBERSHIP-SET PARAMETER ESTIMATION

1.1 Introduction

System identification deals with the formulation of mathematical models of dynamical systems based on input and output data records. The formulation of a model is usually done in two stages [Ljung, 1987]. In the first stage, if the dynamics of the unknown system are known, then a set of candidate models (the model set) can be specified. In other cases, standard model sets (black box models) can be used without reference to the actual system dynamics. The model set itself, is just a mathematical relationship between the system variables. It could be in continuous time or discrete time. It could also be characterized as being linear or nonlinear, deterministic or stochastic. The model set usually contains unknown quantities, which are termed the unknown parameters of the model. The next stage of system identification uses the input-output record and other information to obtain the 'best' model from the model set by choosing the unknown parameters appropriately. This stage of identification has been termed parameter estimation in the literature. In the classical approach, once the system model (with an unknown parameter vector θ^*) has been formulated, then a predictor model (with an adjustable parameter vector θ) is formed, which, yields at every instant of time, a prediction of the system output, based on past information. The predicted output is a function of the parameter vector θ.Then, a criterion of fit is chosen. Perhaps the most widely used criterion is the mean squared prediction error criterion, where the prediction error is the error between the system output and the predictor model output. Once a criterion of fit is chosen, the parameter estimate is chosen to be that value of θ which best fits the criterion. Thus least-squares parameter estimates minimize the mean or average squared prediction error.

System identification forms the core of most adaptive signal processing and adaptive control techniques. In telephony, for example, echo cancellation is required for long distance speech communication [Messerschmitt, 1984]. In the transmission of speech, echos arise primarily due to leakage through the far end hybrid. Speech echo cancellers usually model the hybrid as a FIR filter and adjust the parameters of the predictor filter (also a FIR filter) to minimize the mean-squared error between the echo and the predictor filter output. In high resolution spectral estimation, the signal of interest is usually assumed to be a sum of sinusoids in noise. This signal is often modeled as the output of an IIR filter driven by white noise. The parameters of the filter are then estimated and used to construct an estimate of the spectrum. In adaptive control, a model of the unknown plant is first formed. A common model used is the ARX model described in the next section. The estimated model parameters are then used by the controller to generate the control signal.

Most system model sets incorporate a disturbance term which can represent observation noise or modeling uncertainty. This noise term is usually assumed to be a stochastic process. Some statistical estimation schemes such as maximum likelihood estimation require precise knowledge of the probability density function of the noise. The simpler least-squares schemes require the noise to be white in order to obtain unbiased parameter estimates. If the noise is not white, then unbiased estimates can be obtained by modeling the noise term as a linear regression process. This approach is used in extended least-squares(ELS), generalized least-squares (GLS), recursive maximum likelihood (RML) [Ljung, 1983], output error methods [Goodwin, 1984]. The convergence analysis of all these methods, however, does require that the noise be a stationary stochastic process.

As opposed to classical approaches to parameter estimation which yield point estimates of parameters by optimizing some criterion of fit, membership-set parameter estimation (MSPE) algorithms yield a set of parameter vectors which is compatible with the model structure, observation record and noise constraints. In general, no knowledge of the statistics of the noise process is assumed. However, the noise is assumed to be constrained in some other way, e.g., with bounded energy [Fogel, 1979] or bounded magnitude [Fogel, 1982]. Membership-set algorithms are thus preferable when the noise is too structured as in the case of error occurring when a large order system is modeled by a lower order model. MSPE algorithms yield 100% confidence intervals for the parameter estimates at every time instant. In contrast, confidence intervals for least-squares parameter estimates can be obtained only asymptotically in most cases. Another important feature of recursive MSPE algorithms is a discerning update strategy whereby only a fraction of the incoming data points need be used to construct the membership sets. This not only reduces the total processing time but also enhances the potential for using these algorithms in multi-channel environments. For every observation which is processed, recursive MSPE algorithms can also indicate if the observation is consistent with the model and noise bounds. Thus, the presence of outliers or large modeling inaccuracy can be detected quite easily, in contrast to other parameter estimation schemes.

In this report, attention will be focussed on the estimation of membership sets of parameters of linear discrete time difference equation models. In the remainder of this chapter, an overview of some membership-set algorithms will be provided. In Chapter II, a particular class of membership-set algorithms called optimal bounding ellipsoidal (OBE) estimation algorithms is studied at some length. The algorithms in this class obtain, recursively, ellipsoidal outer bounds of the membership sets of parameters. Though the membership-sets obtained are often larger than those obtained by using other algorithms, the bounding ellipsoid algorithms have the advantages of low computational complexity, analytical tractability, and robustness to parameter variations and finite precision effects. In Chapter III, a particular bounding ellipsoid algorithm, i.e., the DHOBE algorithm of Dasgupta and Huang [Dasgupta, 1987], is used for the estimation of parameters of systems with unobservable inputs. The performance of the algorithm is analyzed and sufficient conditions for satisfactory behavior are derived. In Chapter IV, finite precision effects in the DHOBE algorithm are studied. It is shown that the algorithm remains stable (algorithmic variables remain bounded) in finite word-length environments. Simulation results show that the performance of the algorithm is superior to the recursive least-squares algorithm when the word-length is small. The tracking property of the DHOBE algorithm is studied in Chapter V. Conditions under which the algorithms can track variations in parameters are derived. Modifications to the algorithm are suggested which improve the tracking at the expense of increasing the size of the outer bounding ellipsoidal approximation to the membership set.

The overview of membership-set parameter estimation algorithms commences with an enumeration of the different model structures which are commonly employed in parameter estimation.

1.2 Model Structures

The various types of model structures which are considered in this report are defined below.

MA model: A moving average model is defined by

$$y(t) = b_0u(t) + b_1u(t-1) + ... + b_m u(t-n) + v(t) \tag{1.2.1}$$

where y(t) and u(t) denote the system output and input at time instant t, respectively, and v(t) denotes the disturbance at time t.

AR model: An autoregressive model is defined by the following difference equation

$$y(t) = a_1 y(t-1) + a_2\, y(t-2) + \ldots + a_n\, y(t-n) + v(t) \qquad (1.2.2)$$

where y(t) and v(t) are as defined above.

ARX model: An autoregressive with exogenous input model is defined by

$$y(t) = a_1 y(t-1) + a_2\, y(t-2) + \ldots + a_n\, y(t-n) + b_0 u(t) + b_1 u(t-1) + \ldots + b_m\, u(t-m) + v(t) \qquad (1.2.3)$$

where y(t), u(t) and v(t) are defined above.

ARMA model An autoregressive moving average model is defined by

$$y(t) = a_1 y(t-1) + a_2\, y(t-2) + \ldots + a_n\, y(t-n) + w(t) + c_1 w(t-1) + \ldots + c_r\, w(t-r) \qquad (1.2.4)$$

where w(t) is a unobservable *input or noise sequence and y(t) is the system output.*

In the statistical literature, v(t) and w(t) are assumed to be zero mean, white stochastic processes. However, since the approach used in this report is deterministic, these restrictions will not be imposed. It will be assumed, instead, that v(t) and w(t) are bounded in magnitude. This assumption is quite realistic in practice. For example, observation noise which arises due to quantization or round-off error is bounded, as is the error in measurements due to climatic effects. Similarly, the error due to model mismatch is often bounded.

1.3 Membership Sets

Definition: (Membership set)

Membership set: A membership set is a region in the parameter space which contains all the parameter vectors consistent with the model structure, observation record and noise bounds.

The ARX model which was defined in the last section is a commonly used model structure, since it can model linear time-invariant systems which have poles and zeros. The ARX system model can be expressed as

$$y(t) = \theta^{*T}\Phi(t) + v(t) \tag{1.3.1}$$

with

$$\theta^* = [a_1, a_2, \ldots, a_n, b_0, b_1, \ldots, b_m]^T \tag{1.3.2}$$

and

$$\Phi(t) = [y(t-1), y(t-2), \ldots, y(t-n), u(t), u(t-1), \ldots, u(t-m)]^T \tag{1.3.3}$$

It is assumed that the noise $\{v(t)\}$ is constrained in some way. For simplicity, it is assumed that $\{v(t)\}$ is bounded in magnitude. Hence there exists a known positive γ, such that

$$|v(t)| \le \gamma \tag{1.3.4}$$

Given $\{y(t), u(t), t = 1, 2, \ldots, T)$, the goal of membership-set parameter estimation is to find the smallest set of parameter vectors $\psi(T)$ in euclidean space $\mathbf{R}^N$, where $N = n+m+1$, which is consistent with all the above equations (1.3.1) to (1.3.4).

From (1.3.1) and (1.3.4), it follows that

$$|y(t) - \theta^{*T}\Phi(t)| \le \gamma \tag{1.3.5}$$

So the set S_t enclosed by the hyperplanes

$$H_1(t) = \{ \theta \in \mathbf{R}^N : y(t) - \theta^T\Phi(t) = +\gamma \} \text{ and } H_2(t) = \{\theta \in \mathbf{R}^N : y(t) - \theta^T\Phi(t) = -\gamma \}$$

contains the true parameter. The S_t can be described as the intersection of two half spaces. Given the observation record up to time instant T, the smallest possible membership set will be

$$\psi(T) = \bigcap_{i=1}^{T} S_t$$

Finding a description of the exact membership set is an arduous task, since it may have thousands of vertices for even small values of the order N and data record size T. For example, when N=3, the number of vertices could be as large as T(T-1)+2 [Coxeter, 1973]. Thus for T=100, the membership set could have almost 10,000 vertices. Consequently most MSPE algorithms attempt to obtain a region of simpler shape like an ellipsoid or a box which contains the true membership set $\psi(T)$. Recently, however, two exact polytope bounding algorithms have been proposed [Walter, 1987; Mo, 1988], which recursively yield an exact description of the true membership set. The next section discusses the exact polytope updating algorithm of Mo and Norton and the exact cone updating algorithm of Walter and Lahanier.

1.4 Exact Polytope Bounding

Even though theoretically, the true membership set may have thousands of vertices as the number of data points processed increases, in practice, once the intersection of the first few half spaces is formed, the membership set becomes quite small and so only a small fraction of the incoming half spaces affect the membership set. Hence the number of vertices of the membership set increases quite slowly as the number of data points increases. This provides the motivation for developing algorithms which identify the precise shape of the membership set.

For the membership set at any instant t, the exact polytope updating algorithm (e.p.u. algorithm) of [Mo, 1988], stores a list of all the vertices in terms of their components. For each vertex, a list of all adjacent vertices (the vertex-vertex list), and the hyperplanes which intersect and form the vertex (the vertex-plane list) are also stored. When a new set of parameter bounds (the set S_t) arrives, a test is made to see how S_t intersects the existing membership set $\psi(t-1)$. If the intersection is void, this indicates that either the model structure or noise bound is incorrect. If S_t contains $\psi(t-1)$, then $\psi(t) = \psi(t-1)$, and the parameter bounds provided by S_t are redundant. If the intersection is not void and if S_t does not contain $\psi(t-1)$, then $\psi(t-1)$ has to be updated. This involves (i) calculating the new vertices formed when $H_1(t)$ and/or $H_2(t)$ cut $\psi(t-1)$, and creating vertex-vertex and vertex-plane adjacency lists for the newly formed vertices; (ii) updating the vertex-vertex and vertex-plane lists of the old vertices which belong to $\psi(t-1) \cap S_t$, and (iii) removing the vertices made redundant by S_t.

The exact cone updating (e.c.u.) algorithm of Walter and Lahanier is similar in many ways to the e.p.u. algorithm. It transforms the membership set, which is a convex polyhedron in $\mathbf{R}^N$, into a polyhedral cone in $\mathbf{R}^{N+1}$. Thus the membership set at time t-1 is represented by a cone C_{t-1}. The extreme rays of C_{t-1} are stored as columns of a matrix M. When the parameter bounds due to the t'th observation are applied, then as before, three situations can arise. Either $C_{t-1} \cap S_t = \phi$, or $S_t \supset C_{t-1}$, or one of the hyperplanes $H_1(t)$ or $H_2(t)$ cut C_{t-1}. In the latter case, the extreme rays of C_t will be those of C_{t-1} lying in the set S_t, and new rays belonging to $S_t \cap C_{t-1}$. Each of these new rays is obtained as a linear combination of two adjacent extreme rays of C_{t-1} lying on either side of the cutting hyperplane. Finally, constraints that become redundant at this stage are eliminated.

The computational complexity of both these methods is quite large, even for small order models. The order of complexity at each iteration also grows slowly as new data points are processed, since the number of vertices can increase with every new sample. For these reasons, they are probably not suitable for real time processing applications. It is claimed in [Mo, 1988], that the e.p.u algorithm is superior to the e.c.u. algorithm in terms of numerical robustness. Numerical results have only been reported for the e.p.u algorithm, applied to one ARX parameter estimation problem, and it remains to be seen, how well the algorithms perform in practice.

1.5 Orthotope Bounding

A popular approach in membership-set parameter estimation is to approximate the membership set by an orthotope (a rectangular box) which contains the membership set. This has the added advantage of giving accurate uncertainty intervals for each of the unknown parameters. A membership-set description in terms of parametric uncertainty intervals (PUI's) on the individual parameters, is often preferable to a description of an extremely complicated N-dimensional region. This is particularly true when the parameters have a direct physical interpretation. It has been shown, [Milanese, 1982], that the tightest possible PUI's can be obtained by linear programming. Specifically, Milanese and Belforte showed that the problem can be reduced to solving 2N linear programming problems in N variables with 2T constraints, where N is the order of the model and T is the number of data points. Their algorithm, which is termed the minimum uncertainty interval correct estimator (MUICE), is not recursive and computationally intensive. Consequently, it is not suited for real time applications or to the analysis of non-stationary data. A simple recursive algorithm which constructs outer bounding orthotopes has been proposed [Huang, 1980]. However this algorithm does not yield small enough orthotopes for large order systems. Another recursive algorithm which constructs outer bounding orthotopes has been proposed recently [Pearson, 1986]. The

PUI's obtained by using the algorithm are not as tight as the PUI's of the MUICE, although, they are much easier to evaluate.

It is clear from the discussion above, that both the exact polytope bounding and the orthotope bounding algorithms involve considerable amounts of computation. In fact, the computational complexity of the exact polytope bounding algorithms depends heavily on the characteristics of the data. Thus these algorithms are not suited for real time applications. In the next chapter, several ellipsoidal bounding algorithms will be discussed, all of which have the advantageous features of low computational complexity and analytical tractability. The latter feature has simplified the application of the algorithms to different cases such as ARMA parameter estimation and output error parameter estimation, and, has enabled its implementation via lattice filters and systolic arrays. All this and much more coming up, so don't go away!

CHAPTER II

BOUNDING ELLIPSOIDAL PARAMETER ESTIMATION

2.1 Introduction

The optimal bounding ellipsoid algorithms outer bound the membership set of parameters by ellipsoids in the parameter space. The idea of using ellipsoids to bound sets was originally proposed by Schweppe [Schweppe, 1967], in the context of state estimation. He formulated a recursive algorithm for completely specified dynamic systems, with unknown but bounded inputs and bounded observation errors. At every instant, the algorithm yields ellipsoidal sets, which contain the true time varying system state. The state estimate can be taken to be the center of the ellipsoid. The algorithm differs from the Kalman filter, developed for linear dynamical systems with gaussian inputs and noise, only in the gain sequence. Following Schweppe, Fogel proposed a recursive algorithm for calculating ellipsoidal outer bounds of the membership sets of parameters, assuming energy constraints on the noise [Fogel, 1979]. By imposing instantaneous bounds on the noise, Fogel and Huang [Fogel, 1982] came up with membership-set algorithms, wherein, the size of the bounding ellipsoids is optimized according to different criteria. A by-product of the optimization procedure is a discerning update strategy which makes efficient use of data [Fogel, 1982]. Based on their work, different bounding ellipsoidal algorithms have been proposed in the past few years. In this chapter, the optimal bounding ellipsoid (OBE) algorithms of [Fogel, 1982] are presented first. Then in Section 2.3, a more recent OBE algorithm, the DHOBE algorithm [Dasgupta, 1987], which uses a slightly different ellipsoidal formulation and optimization criterion, is discussed at some length. An analogy between weighted least-squares and the OBE algorithms is drawn in Section 2.4. Some simulated situations in which the performance of the OBE algorithms is seen to be markedly superior to the recursive least-squares algorithm are presented in Section 2.5. Finally, in Section 2.6, an improvement to the OBE algorithms of Fogel and Huang [Belforte, 1985] is discussed and the improved algorithm is compared with the other OBE algorithms via simulations.

2.2 The OBE Algorithms

As mentioned earlier, the OBE algorithms of Fogel and Huang [Fogel, 1982], recursively obtain ellipsoidal outer bounds to the membership set. The model structure considered is the ARX model of (1.3.1)

$$y(t) = \theta^{*T}\Phi(t) + v(t) \tag{2.2.1}$$

where θ^*, the true parameter vector, and, $\Phi(t)$, the regressor vector, are N dimensional vectors given by (1.3.2) and (1.3.3) respectively. The noise $v(t)$ is assumed to be uniformly bounded in magnitude

$$| v(t) | \leq \gamma \tag{2.2.2}$$

In order to develop a recursive formulation for the bounding ellipsoids, it is assumed that at time instant t-1, the true membership set is outer bounded by the ellipsoid E_{t-1} described by

$$E_{t-1} = \{\theta \in \mathbf{R}^N : [\theta - \theta(t-1)]^T P^{-1}(t-1) [\theta - \theta(t-1)] \leq 1 \} \tag{2.2.3}$$

where $P^{-1}(t-1)$ is a positive definite matrix, and $\theta(t-1)$ is the center of the ellipsoid. At time instant t, observation $y(t)$, defines a set S_t, which is a degenerate ellipsoid in the parameter space

$$S_t = \{\theta \in \mathbf{R}^N : [y(t) - \theta^T\Phi(t)]^2 \leq \gamma^2 \} \tag{2.2.4}$$

From (2.2.1) and (2.2.2) it is clear that S_t contains the true parameter vector. An ellipsoid E_t which contains the intersection of E_{t-1} and S_t is then given by

$$E_t = \{\theta \in \mathbf{R}^N : [\theta - \theta(t-1)]^T P^{-1}(t-1) [\theta - \theta(t-1)] + \lambda_t [y(t) - \theta^T\Phi(t)]^2 \leq 1 + \lambda_t \gamma^2\} \qquad (2.2.5)$$

where λ_t is a positive time varying updating gain which is chosen to minimize the size of the ellipsoid E_t. By performing some tedious, but straightforward algebraic manipulation, it can be shown that E_t can be expressed in the form

$$E_t = \{\theta \in \mathbf{R}^N : [\theta - \theta(t)]^T P^{-1}(t) [\theta - \theta(t)] \leq 1\} \qquad (2.2.6)$$

where

$$Q^{-1}(t) = P^{-1}(t-1) + \lambda_t \Phi(t)\Phi^T(t) \qquad (2.2.7)$$

$$P^{-1}(t) = \sigma^2(t) Q^{-1}(t) \qquad (2.2.8)$$

$$\sigma^2(t) = 1 + \lambda_t \gamma^2 - \frac{\lambda_t [y(t)-\Phi^T(t)\theta(t-1)]^2}{1+\lambda_t \Phi^T(t)P(t-1)\Phi^T(t)} \qquad (2.2.9)$$

$$\theta(t) = \theta(t-1) + \lambda_t Q(t)\Phi(t)[y(t)-\Phi^T(t)\theta(t-1)] \qquad (2.2.10)$$

The matrix inversion required in (2.2.10) can be circumvented by using the matrix inversion lemma in (2.2.7), which yields

$$Q(t) = P(t-1) - \lambda_t \frac{P(t-1)\Phi(t)\Phi^T(t)P(t-1)}{1+\lambda_t \Phi^T(t)P(t-1)\Phi^T(t)} \qquad (2.2.11)$$

In order to ensure that the initial ellipsoid E_0 contains θ^*, E_0 is taken to be a large ball centered around zero, i.e. $P(0) = M.\mathbf{I}$, where M is a large number and $\mathbf{I}$ is the N by N identity matrix, and $\theta(0) = 0$. If an initial estimate of $\theta^*$is available, then $\theta(0)$ could be set to that value.

In the minimal volume sequential (MVS) algorithm, at every time instant t, the determinant of P(t), which is proportional to the volume of the enclosing ellipsoid E_t, is minimized with respect to λ_t. This yields the following formula for the gain factor

If $2N[\gamma^2 - \delta^2(t)] - G(t) \geq 0$, then $\lambda_t = 0$ (2.2.12a)

with $\delta(t) = y(t) - \Phi^T(t)\theta(t-1)$

otherwise

$$\lambda_t = \frac{-\alpha_2 + \sqrt{\alpha_2^2 - 4\alpha_1\alpha_3}}{2\alpha_1} \tag{2.2.12b}$$

where

$$\alpha_1 = (2N-1)\gamma^2 G^2(t) \tag{2.2.12c}$$

$$\alpha_2 = G(t)[(4N-1)\gamma^2 - G(t) + \delta^2(t)] \tag{2.2.12d}$$

$$\alpha_3 = 2N[\gamma^2 - \delta^2(t)] - G(t) \tag{2.2.12e}$$

and

$$G(t) = \Phi^T(t)P(t-1)\Phi^T(t)) \tag{2.2.12f}$$

Note that when $\lambda_t = 0$, $E_t = E_{t-1}$, i.e. $\theta(t) = \theta(t-1)$ and $P(t) = P(t-1)$. The evaluation of λ_t is thus the basis of a discerning update strategy, whereby, the " innovativeness " of the observation pair $\{y(t), \Phi(t)\}$ is checked in (2.2.12a). An update ($\lambda_t \neq 0$) occurs only if it is possible to construct an ellipsoid E_t, which bounds the intersection of E_{t-1} and S_t, and whose volume is less than E_{t-1}.

The computational complexity of the ellipsoid updating formula (2.2.8)-(2.2.11) is $O(N^2)$, which is same order of computational complexity as that of the RLS algorithm. For the discerning update strategy , the major contributor to the computational cost is the computation of $G(t) = \Phi^T(t)P(t-1)\Phi^T(t)$, which takes $(N^2 + N)$ multiplications and $(N+1)(N-1)$ additions.

Instead of choosing λ_t to minimize the volume, it can be chosen to minimize the sum of the semi-axes of the bounding ellipsoid E_t. This is achieved by minimizing the trace of the matrix P(t). The resulting minimum trace sequential (MTS) algorithm has a different update strategy . In case of an update, in order to find the optimum updating gain factor, the positive root of a certain third order polynomial has to be found [Fogel, 1982]. The MTS algorithm, therefore, has a higher computational cost than the MVS algorithm.

2.3 The DHOBE Algorithm

The updating gain factors of the MVS and MTS algorithms of the above section are chosen to minimize the size of the bounding ellipsoid. This is no doubt desirable, when the parameters of the unknown system are fixed. However, if in case, the true parameter changes after the the ellipsoid has shrunk, it is possible that the resulting bounding ellipsoid, if it exists, will no longer contain the true parameter and hence it will not be possible to track the true parameter. Thus from the point of view of tracking time varying parameters, it may not always be advantageous to minimize the size of the bounding ellipsoids.

The motivation for the development of the DHOBE algorithm, stems more from the point of view of adaptive filtering and prediction error minimization, rather than from membership-set parameter estimation. This accounts for the similarity between the DHOBE algorithm and some bounded error algorithms proposed in the adaptive control literature [Fortescue, 1981; Ortega, 1987]. The quantity which is minimized in the DHOBE algorithm is a certain upper bound on the normalized parameter estimation error. This yields an updating rule which has a computational complexity of only (N+1) multiplies. Furthermore, minimizing with the above criterion greatly enhances the analytical tractability of the algorithm. Analysis shows that the *a priori* prediction error is asymptotically bounded by the bound on the noise. Additionally, a degree of uncorrelatedness between the inputs and the noise is sufficient for asymptotic cessation of updates in the fixed parameter case. The updating gain factor in the algorithm also plays the dual role of a forgetting factor. This improves the tracking capability of the algorithm *vis a vis* any of the MSPE algorithms. On the flip side, since the size of the bounding ellipsoids is no longer the criterion for optimization, the DHOBE algorithm, in general, yields bounding ellipsoids which are larger than those yielded by other OBE algorithms. The above remarks will become clearer after the following discussion.

The sequence of optimal bounding ellipsoids in the DHOBE algorithm is developed as follows. Let the bounding ellipsoid at time instant t-1 be

$$E_{t-1} = \{\theta \in \mathbf{R}^N : [\theta - \theta(t-1)]^T P^{-1}(t-1) [\theta - \theta(t-1)] \leq \sigma^2(t-1) \} \qquad (2.3.1)$$

where, as in the previous section, $P^{-1}(t-1)$ is a positive definite matrix, and $\theta(t-1)$ is the center of the ellipsoid. The factor $\sigma^2(t-1)$ is a positive time varying scalar, which along with $P^{-1}(t-1)$ determines the size of E_{t-1}. Since the true parameter $\theta^* \in E_{t-1}$, $\sigma^2(t-1)$ can also be thought of as being an upper bound on the normalized parameter estimation error

$$V(t-1) = [\theta^* - \theta(t-1)]^T P^{-1}(t-1) [\theta^* - \theta(t-1)]$$

An ellipsoid E_t which bounds the intersection of E_{t-1} and the set S_t, where S_t is described by (2.2.4), is then given by

$$E_t = \{\theta \in \mathbf{R}^N : (1-\lambda_t)[\theta - \theta(t-1)]^T P^{-1}(t-1) [\theta - \theta(t-1)] + \lambda_t [y(t) - \theta^T \Phi(t)]^2 \leq (1-\lambda_t)\sigma^2(t-1) + \lambda_t \gamma^2 \} \qquad (2.3.2)$$

Comparing (2.2.5) and (2.3.2), it can be seen that in (2.3.2), λ_t is an updating gain factor and $(1-\lambda_t)$ is a forgetting factor. By performing some algebraic manipulations, an expression for E_t can be obtained as

$$E_t = \{\theta \in \mathbf{R}^N : [\theta - \theta(t)]^T P^{-1}(t) [\theta - \theta(t)] \leq \sigma^2(t) \} \tag{2.3.3}$$

where

$$P^{-1}(t) = (1 - \lambda_t)P^{-1}(t-1) + \lambda_t \Phi(t)\Phi^T(t) \tag{2.3.4}$$

$$\sigma^2(t) = (1-\lambda_t)\,\sigma^2(t-1) + \lambda_t \gamma^2 - \frac{\lambda_t (1-\lambda_t) [y(t) - \Phi^T(t)\theta(t-1)]^2}{1-\lambda_t + \lambda_t \Phi^T(t)P(t-1)\Phi^T(t)} \tag{2.3.5}$$

$$\theta(t) = \theta(t-1) + \lambda_t P(t)\Phi(t)[y(t) - \Phi^T(t)\theta(t-1)] \tag{2.3.6}$$

Using the matrix inversion lemma in (2.3.4) yields

$$P(t) = \frac{1}{1-\lambda_t}[P(t-1) - \frac{\lambda_t P(t-1)\Phi(t)\Phi^T(t)P(t-1)}{1-\lambda_t + \lambda_t \Phi^T(t)P(t-1)\Phi^T(t)}] \tag{2.3.7}$$

The initial conditions are chosen to ensure that $\theta^* \in E_0$. A possible choice is

$$P(0) = \mathbf{I}, \text{ and } \sigma^2(0) = 1/\varepsilon^2 \text{ where } \varepsilon << 1.$$

The updating gain λ_t is chosen to minimize $\sigma^2(t)$ at every instant t. The minimization procedure yields the following updating criterion

$$\text{If} \quad \sigma^2(t-1) + \delta^2(t) \leq \gamma^2 \quad \text{then} \quad \lambda_t = 0 \text{ (No update)} \tag{2.3.8}$$

where the *a priori* prediction error

$$\delta(t) = y(t) - \Phi^T(t)\theta(t-1) \tag{2.3.9}$$

Otherwise if $\sigma^2(t-1) + \delta^2(t) > \gamma^2$, then the optimum value of λ_t is non-zero and calculated according to

$$\lambda_t = \min(\alpha, \nu_t) \tag{2.3.10}$$

where

$$\nu_t = \begin{cases} \alpha & \text{if } \delta^2(t) = 0 \quad (2.3.11) \\ \dfrac{1-\beta(t)}{2} & \text{if } G(t) = 1 \quad (2.3.12) \\ \dfrac{1}{1-G(t)}\left[1-\sqrt{\dfrac{G(t)}{1+\beta(t)[G(t)-1]}}\right] & \text{if } 1+\beta(t)[G(t)-1] > 0 \quad (2.3.13) \\ \alpha & \text{if } 1+\beta(t)[G(t)-1] \le 0 \quad (2.3.14) \end{cases}$$

where α is a user chosen upper bound on $\hat{\lambda}_t$ satisfying

$$0 < \alpha < 1 \tag{2.3.15}$$

and

$$G(t) = \Phi^T(t)P(t-1)\Phi^T(t) \tag{2.3.16}$$

and

$$\beta(t) = \frac{\gamma^2 - \sigma^2(t-1)}{\delta^2(t)} \tag{2.3.17}$$

The main results of the convergence analysis of the DHOBE algorithm [Dasgupta. 1987] are

(i) $\{\sigma^2(t)\}$ is a non-increasing sequence and

$$\lim_{t\to\infty} \sigma^2(t) = \sigma^2, \text{ where } 0 \le \sigma^2 \le \gamma^2$$

(ii) The eigenvalues of P(t) are upper and lower bounded provided the inputs are sufficiently rich in frequency and sufficiently uncorrelated with the noise. i.e. if there exist $\beta_1, \beta_2 > 0$, such that for some M and all t

$$\beta_1 \mathbf{I} \le \sum_{i=t+n}^{t+M} W(t)W^T(t) \le \beta_2 \mathbf{I} \tag{2.3.18}$$

where

$$W(t) = [u(t), u(t-1),..,u(t-n-m), v(t), v(t-1), ..,v(t-n)]^T,$$

with n being the order of the AR part, and m being the order of the exogenous part; then there exist $\alpha_1, \alpha_2 > 0$, such that for all t

$$\alpha_1 \mathbf{I} \le P(t) \le \alpha_2 \mathbf{I}$$

(iii) If G(t) is bounded, i.e if the conditions of (2.3.18) hold then,

(a) The a priori prediction errors are bounded

$$\delta^2(t) \rightarrow [0, \gamma^2]$$

and

(b) The parameter estimate difference converges to zero, i.e.

$$\| \theta(t) - \theta(t-1) \| \rightarrow 0$$

(iv) If the conditions of (2.3.18) hold and θ^* is constant, then

$$\lim_{t\rightarrow\infty} \lambda_t \rightarrow 0$$

A detailed derivation of these convergence results and the performance of the algorithm in computer simulations can be found in [Dasgupta, 1987].

Looking at the update equations of the OBE algorithms, it is clear that the algorithms are similar to the weighted recursive least-squares (WRLS) algorithm. The next section shows that the OBE algorithms are in fact special cases of the WRLS algorithm.

2.4 Weighted Least-Squares and the OBE Algorithms

The conventional weighted least squares algorithm [Ljung, 1983] minimizes the criterion

$$V_T(\theta) = \frac{1}{T}\sum_{i=1}^{T}\alpha_i [y(i) - \theta^T\Phi(i)]^2 \qquad (2.4.1)$$

The need for weighting the observations can arise if, for example, the noise variances are time varying and in that case, α_t can be taken to be inversely proportional to the variance of the noise. In fact, this choice of α_t is optimal when the noise is uncorrelated and gaussian, i.e., the resulting parameter estimates have the minimum covariance over all possible choices of the forgetting factor. If in addition, it is required to discount older values, then the following criterion can be used

$$V'_T(\theta) = \frac{1}{T}\sum_{i=1}^{T} \beta(T,i)[y(i) - \theta^T\Phi(i)]^2 \tag{2.4.2}$$

where

$$\beta(T,i) = \lambda(T)\beta(T-1,i), \quad 1 \le i \le T-1 \tag{2.4.3}$$

which can also be written as

$$\beta(T,i) = [\prod_{j=i+1}^{T} \lambda(j)]\alpha_i \text{ , with } \beta(i,i) = \alpha_i \tag{2.4.4}$$

The value of θ which minimizes $V'_T(\theta)$, can easily be computed

$$\theta(T) = \left[\sum_{i=1}^{T} \beta(T,i)\Phi(i)\Phi^T(i)\right]^{-1}\left[\sum_{i=1}^{T} \beta(T,i)\Phi(i)y(i)\right] \tag{2.4.5}$$

A recursive form for $\theta(t)$, can be easily developed by using (2.4.3) and (2.4.5)

$$\theta(t) = \theta(t-1) + \alpha_t P(t)\Phi(t)[y(t)-\Phi^T(t)\theta(t-1)] \tag{2.4.6}$$

where

$$P^{-1}(t) = \lambda(t) P^{-1}(t-1) + \alpha_t\Phi(t)\Phi^T(t) \tag{2.4.7}$$

Thus the parameter vector $\theta(t)$ described by (2.4.6) and (2.4.7) minimizes $V'_t(\theta)$. Comparing (2.4.6) and (2.4.7) with (2.3.4) and (2.3.6) of the DHOBE algorithm, it is apparent that λ_t in (2.3.6) plays the same role as α_t in (2.4.6) and (2.4.7). Also $(1- \lambda_t)$

in (2.3.4) is equivalent to $\lambda(t)$ in (2.4.7). Thus the DHOBE algorithm minimizes the criterion

$$V_{OBE}(\theta) = \frac{1}{T}\sum_{i=1}^{T} q_{i,T}[y(i) - \theta^T\Phi(i)]^2 \qquad (2.4.8)$$

where

$$q_{i,T} = [\prod_{j=i+1}^{T}(1-\lambda_j)]\,\lambda_i \text{ , and } q_{i,i} = \lambda_i \qquad (2.4.9)$$

Thus the DHOBE algorithm can be described as a weighted recursive least-squares with forgetting factor algorithm, with the weight α_t and forgetting factor $\lambda(t)$ given by

$$\lambda(t) = 1 - \lambda_t \text{ , and } \alpha_t = \lambda_t$$

Alternatively, the DHOBE algorithm can be described as a weighted recursive least-squares algorithm with non-causal weights

$$\alpha_t = q_{t,T}$$

For the MVS or MTS OBE algorithms, it is easy to show that the corresponding relations are

$$\lambda(t) = \frac{1}{\sigma^2(t)} \text{ and } \alpha_t = \frac{\lambda_t}{\sigma^2(t)}$$

Thus these algorithms are also special cases of the WRLS with forgetting factor algorithm.

One implication of such a relation between the OBE algorithms and the WRLS algorithms is that it facilitates application of many of the ideas and concepts from the least-squares adaptive filtering literature. For example, it opens up the possibility of developing exact lattice and fast transversal filter implementations of the OBE algorithms. In fact, this equivalence has already been exploited to develop a systolic array implementation of the MVS algorithm [Deller, 1989].

Another implication is that, even though the formulation of the OBE algorithms is motivated by membership-set-theoretic considerations, it turns out that the parameter estimates (the centers of the ellipsoids) minimize the average weighted prediction error. In the MTS and MVS algorithms, the choice of the weighting sequence is motivated by a desire to reduce the size of the bounding ellipsoids. In the DHOBE algorithm, the weighting factor is chosen to minimize an instantaneous upper bound on the normalized parameter estimation error. Due to the different choices of the weighting and forgetting factor sequences, the parameter estimate trajectories of the different OBE algorithms and the popular exponentially weighted least-squares (EWLS) algorithm (with $\lambda(t)=\lambda < 1$, and $\alpha_t = 1$) may differ significantly. To illustrate this point, some simulated examples in which, the parameter estimation error of the OBE algorithms is markedly lower than that of the RLS and the EWLS algorithms are presented next.

2.5 Performance in Sinusoidal and Impulsive Noise

In many situations, the noise process affecting the observations may not be a white noise process, or may not even be stationary. For example, the noise may have a large sinusoidal component, as in the case of observations affected by electromagnetic interference, and in the case of helicopter flight data [Goodwin, 1987]. In some cases, the noise may be bursty or impulsive, and thus highly non-stationary. It is interesting to compare the performance of the OBE algorithms to RLS algorithm in these situations. The first two examples compare the performance of the DHOBE and MVS algorithms to the RLS algorithm when the noise is an additive sum of white noise and a sinusoid.

Example 2.1

The unknown system is described by an ARX(2,2) model

$$y(t) = -0.4\, y(t-1) - 0.85 y(t-2) - 0.2\, u(t) - 0.7 u(t-1) + v(t)$$

The input u(t) is generated by a pseudo-random number generator which is uniformly distributed in [-1,+1]. The noise process is generated by

$$v(t) = A \sin(\omega_0 t) + (1 - A)w(t) \tag{2.5.1}$$

where A, the amplitude of the sinusoid, satisfies $0 \leq A \leq 1$, and w(t) is white and uniformly distributed in [-1,+1]. The frequency $\omega_0=\pi/10$. The amplitude of the sinusoid is varied from 0 to 1 and for each value of A, ten Monte Carlo runs, with 500 points each, of the RLS and OBE algorithms are performed. The estimation error is measured by calculating the final mean-squared parameter estimation error (MSPEE) defined by

$$\text{MSPEE (db)} = 10 \log\left[\frac{1}{N}\sum_{i=1}^{N} \|\theta_i(500) - \theta^*\|^2\right] \tag{2.5.2}$$

Figure 2.1 displays the variation in MSPEE for the RLS and OBE algorithms as A is varied. The mean-square error in the RLS estimates increases drastically as the amplitude of the sinusoidal disturbance increases. This is because the RLS estimates are biased when the noise is correlated [Ljung, 1983]. The estimates of the OBE algorithms appear insensitive to the amplitude of the sinusoid. Unfortunately, one cannot derive any general conclusions about the superiority of the OBE algorithms in the presence of sinusoidal disturbances, since such behavior may be specific to the particular ARX(2,2) system considered.

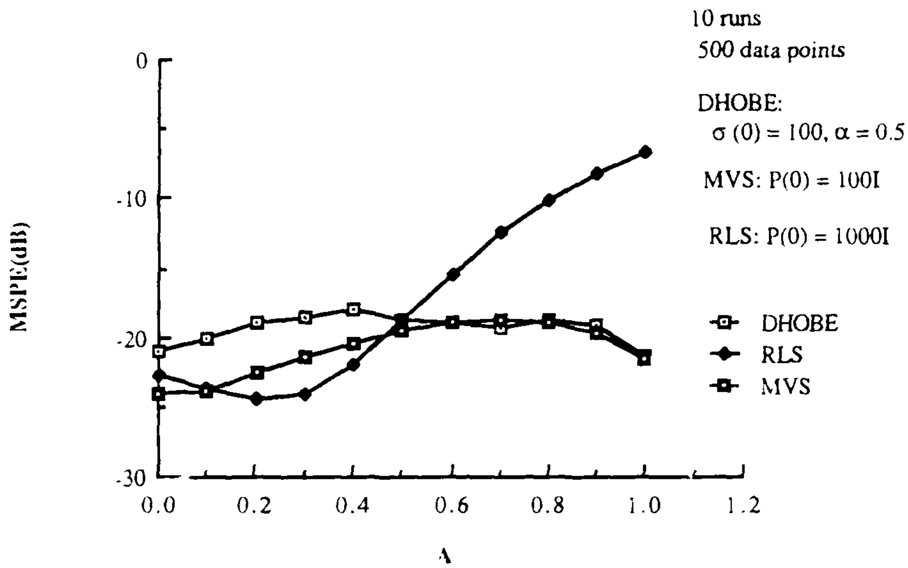

Figure 2.1 Mean-squared parameter estimation error for the OBE and RLS algorithms applied to an ARX(2,2) model with a sinusoidal disturbance

Example 2.2

In order to make some definitive statements about the bias, the behavior of both the algorithms is investigated for a simple ARX(1,1) model.

$$y(t) = x\, y(t-1) + b_0\, u(t) + v(t) \tag{2.5.3}$$

The noise v(t) and the input u(t) is as in Example 2.1. The absolute value of x is required to be less than unity to ensure stability of the system. An expression for the bias in the RLS parameter estimates for this simple model is derived in Appendix 2A. It is shown that if $[a,b\,]^T$ are the values of the parameter estimates that the RLS algorithm yields asymptotically, i.e. these values minimize the mean-squared prediction error, then for the system (2.5.3)

$$a = x + \frac{\frac{A^2}{2}\left[\cos\omega_0 - x\right]}{\frac{A^2}{2} + \left(\sigma_u^2 + (1-A)^2\sigma_w^2\right)\left[\frac{1 - 2x\cos\omega_0 + x^2}{1 - x^2}\right]}$$

and

$$b = b_0 \tag{2.5.4}$$

where σ_w^2 and σ_u^2 are the variances of the white noise w(t) and input u(t) respectively. Thus only the AR estimate is biased and the bias depends on the noise and input variances, the amplitude and frequency of the sinusoid, and the true AR coefficient. In particular, the bias is zero when $A = 0$, and the sign of the bias is the same as the sign of $(\cos\omega_0 - x)$. Unfortunately, a corresponding expression for the bias in the OBE estimates is difficult to derive on account of the presence of the data dependent updating and forgetting factor. In order to get some comparison of the bias possible in the two algorithms, the value of A is now fixed at 1, i.e. the disturbance is a pure sinusoid with $\omega_0 = \pi/10$, and x, the system AR coefficient is varied from +1 to −1. The value of b_0 is set equal to unity, and the input u(t) is generated as in Example 2.1. The asymptotic bias is computed by averaging over 10 Monte Carlo runs of 500 points each. Figure 2.2 and Figure 2.3 show the variation in the bias in the parameter estimates as x is varied. It is clear that the OBE estimates are biased for many values of x. However, the bias in the AR estimate is significantly lower than the bias in the RLS autoregressive estimate. The values of the bias in the RLS algorithm, yielded by simulation are very close to the values predicted by (2.5.4), thus verifying the analysis of Appendex 2A. Unlike the RLS case, the MA estimates yielded by the DHOBE algorithm seem to be slightly biased.

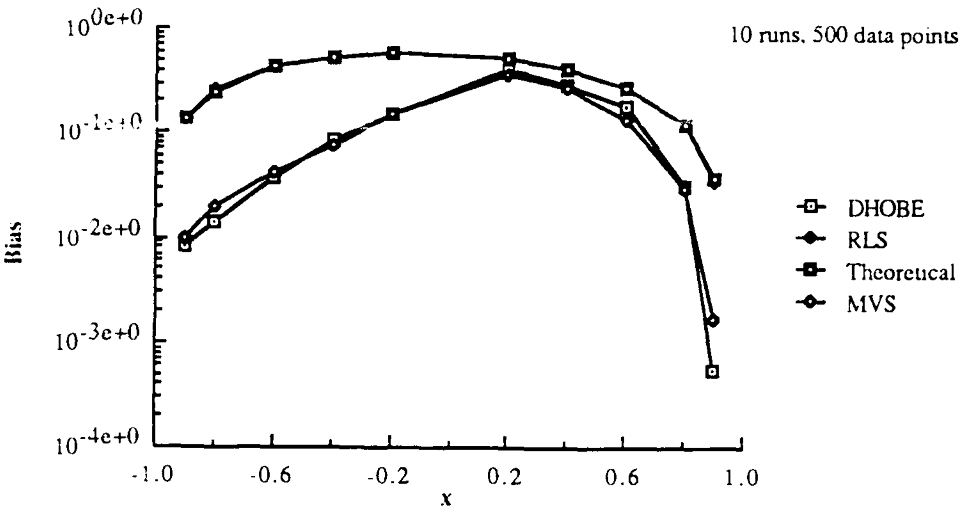

Figure 2.2 Variation in the bias of the RLS and OBE AR parameter estimates as the AR coefficient is varied for an ARX(1,1) system model

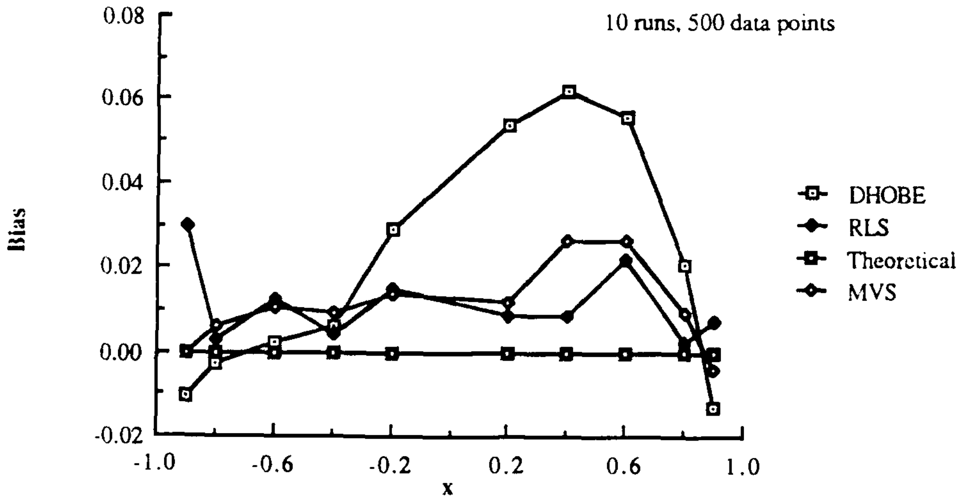

Figure 2.3 Variation in the bias of the RLS and OBE MA parameter estimates

The average final values of $[\sigma^2(t)]^2 \det [P(t)]$, which is a measure of the final volume, and $[\sigma^2(t)] \operatorname{Tr} [P(t)]$, which is measure of the sum of the semi axes, for the DHOBE algorithm, are plotted against x in Figure 2.4. Also plotted are the corresponding values for the MVS algorithm. It appears that the ellipsoids are large at the values of x at which the bias is significant.

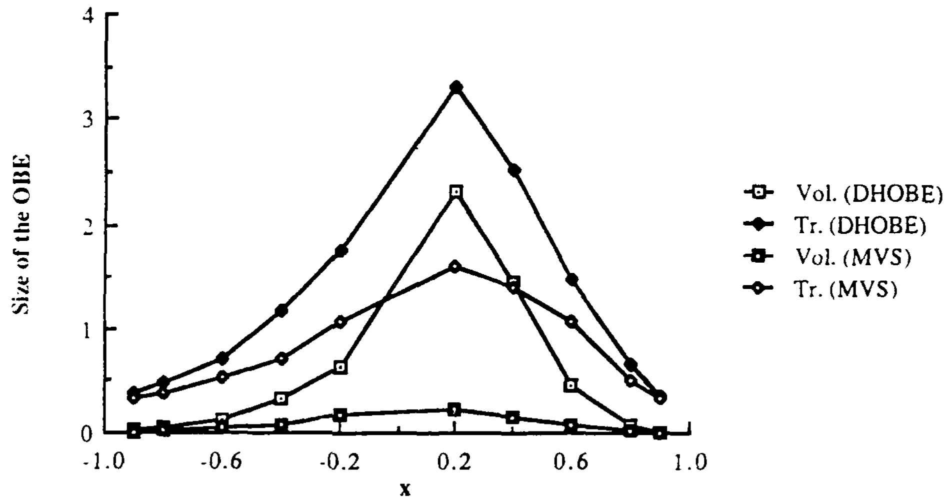

Figure 2.4 Average final volume and sum of the semi-axes for Example 2.2

Example 2.3

In this example the performance of the OBE and RLS algorithms is compared when the observations are affected by bursts of equal amplitude disturbances. The system model considered is the same ARX(2,2) model of Example 2.1.

$$y(t) = -0.4\, y(t-1) - 0.85y(t-2) - 0.2\, u(t) - 0.7u(t-1) + v(t)$$

The noise process $\{v(t)\}$ is now generated as follows. At every instant t, t=1,2,..,1000, a random number $w(t) \in [0,1]$ is generated. If w(t) is greater than 1–*prob* (where *prob* is the preassigned probability of a burst occurring), then v(t+j), j = 0,1,..4, is set equal to unity. Thus the noise sequence is composed of bursts of impulses of burst length ≥ 5. The average parameter estimation error MSPEE, was computed over 10 runs of 1000 points each for the RLS (forgetting factor $\lambda=1$, $P(0) = 1000\mathbf{I}$), EWLS ($P(0) = 1000\mathbf{I}$, $\lambda = 0.99$ and $\lambda = 0.9$), the DHOBE (with $\alpha = 0.5$, $P(0) = \mathbf{I}$, and $\sigma^2(0) = 100$) and the MVS ($P(0) = 100\mathbf{I}$) algorithms. Table 2.1, lists the MSPEE values in dBs for two different values of *prob*. The quantity in parentheses in the first column is the average number of impulsive noise points in the 1000 point data records. The quantities in parentheses in the last two columns are the average final volumes and final sums of semi-axes, respectively, for the DHOBE and MVS algorithms. The performance of both the DHOBE and the MVS algorithm is vastly superior to that of the RLS and EWLS algorithms when the number of bursts is large. However, as *prob* decreases, the performance of the RLS algorithm, becomes comparable, and of course, for very small values of *prob*, the parameter estimation error of the RLS algorithm would be lower than that of the OBE algorithms. It is surprising to see that, in both the cases, the DHOBE algorithm yields ellipsoids of smaller volumes than the MVS algorithm. Since the construction of the bounding ellipsoid E_t from E_{t-1} is different for the MVS and DHOBE algorithms (c.f (2.2.5) and (2.3.2)), the DHOBE algorithm,can in principle, yield ellipsoids with smaller volumes than the minimum volume sequential algorithm.

TABLE 2.1

prob	RLS	EWLS		DHOBE	MVS
		λ=0.99	λ=0.9		
0.05 (239)	-14.9	-14.05	-7.89	-26.9 ($4x10^{-5}$, 0.35)	-28.83 ($1.6x10^{-4}$, 0.52)
0.02 (105)	-20.37	-18.14	-8.6	-22.67 ($8x10^{-4}$, 0.73)	-21.59 ($3.6x10^{-3}$, 1.16)

Discussion of Simulation Results

In the presence of sinusoidal disturbances, the unweighted RLS algorithm yields biased estimates. The estimates of the OBE algorithms are also biased, however, the bias is less than the bias in the unweighted RLS algorithm. The bias in the OBE parameter estimates appears to go hand in hand with an increase in the volume of the bounding ellipsoids. In the impulsive noise case, the mean-square error in the OBE estimates is observed to increase as the number of impulses decrease. This behavior is contradictory to the behavior of the RLS estimates and deserves further investigation.

2.6 A Modification to the MVS Algorithm

It has been observed that the MVS algorithm does not always yield a bounding ellipsoid with minimum volume (the minimum over all ellipsoids formulated according to (2.2.5)). Such a situation can occur when either, (i) one of the hyperplanes- $H_1(t)$ or $H_2(t)$, which define S_t , does not intersect the bounding ellipsoid E_{t-1}; or (ii) both $H_1(t)$ or $H_2(t)$ do not intersect E_{t-1}. In the latter case, the smallest possible E_t is E_{t-1} itself. In the former case, the non-intersecting hyperplane, can be replaced by a parallel hyperplane tangent to E_{t-1}. Then, by appropriately redefining the set S_t, a minimum volume ellipsoid, which bounds the intersection of E_{t-1} and the new S_t can be found, which will have a smaller volume than the ellipsoid obtained by the conventional MVS algorithm. This is the essence of the modification proposed by Belforte and Bona [Belforte, 1985]. The modified algorithm is developed as follows.

The equations defining $H_1(t)$ and $H_2(t)$ are

$$H_1(t) = \{\theta \in \mathbf{R}^N : \ \theta^T\Phi(t) = y(t) - \gamma\}$$

and

$$H_2(t) = \{\theta \in \mathbf{R}^N : \ \theta^T\Phi(t) = y(t) + \gamma\}$$

$H_1(t)$ is the lower hyperplane and $H_2(t)$ is the upper one. Assume that $H_1(t)$ does not intersect E_{t-1}. Then the equation of $H_1'(t)$ parallel to $H_1(t)$ and tangential to E_{t-1} is

$$H_1'(t) = \{\theta \in \mathbf{R}^N : \ \theta^T\Phi(t) = z_1\}$$

z_1 can be found as follows. Assume that $H_1'(t)$ intersects E_{t-1} at θ_0. Since $H_1'(t)$ is orthogonal to the gradient vector at θ_0, for any $\theta \in H_1'(t)$

$$\left\{\frac{\partial}{\partial\theta}\left[(\theta-\theta(t-1))^T P^{-1}(t-1)(\theta-\theta(t-1))\right]\right\}^T_{\theta=\theta_0}(\theta-\theta_0) = 2(\theta_0-\theta(t-1))^T P^{-1}(t-1)(\theta-\theta_0)$$

$$= 0$$

Since $H_1'(t)$ is also orthogonal to $\Phi(t)$ and $\theta-\theta_0$ lies in $H_1'(t)$

$$P^{-1}(t-1)\left(\theta_0 - \theta(t-1)\right) = \eta\Phi(t)$$

for some η, and since θ_0 is on the boundary of E_{t-1},

$$\left(\theta_0 - \theta(t-1)\right)^T P^{-1}(t-1)\left(\theta_0 - \theta(t-1)\right) = \eta^2\Phi^T(t)P(t-1)\Phi(t) = 1$$

Hence the projection of $\theta_0 - \theta(t-1)$ onto $\Phi(t)$ is of length

$$h_t = \frac{\Phi^T(t)\left(\theta_0 - \theta(t-1)\right)}{\sqrt{\Phi^T(t)\Phi(t)}} = \eta\frac{\Phi^T(t)P(t-1)\Phi(t)}{\sqrt{\Phi^T(t)\Phi(t)}} = \frac{\sqrt{\Phi^T(t)P(t-1)\Phi(t)}}{\sqrt{\Phi^T(t)\Phi(t)}}$$

$$= \frac{\sqrt{G(t)}}{\sqrt{\Phi^T(t)\Phi(t)}}$$

where, as before, $G(t) = \Phi^T(t)P(t-1)\Phi(t)$. But, since the equation of a hyperplane parallel to $H_1'(t)$ and passing through $\theta(t-1)$ is $\theta^T\Phi(t) = \theta^T(t-1)\Phi(t)$, therefore

$$h_t = \frac{\theta^T(t-1)\Phi(t) - z_1}{\sqrt{\Phi^T(t)\Phi(t)}}$$

Thus

$$z_1 = \theta^T(t-1)\Phi(t) - \sqrt{G(t)}$$

It can be shown similarly that the equation of a hyperplane parallel to $H_2(t)$ and tangent to E_{t-1} is

$$H_2'(t) = \{\theta \in \mathbf{R}^N : \theta^T\Phi(t) = z_2 = \theta^T(t-1)\Phi(t) + \sqrt{G(t)}\ \}$$

The modified algorithm thus consists of the following steps

(i) Evaluate $z_1 = \theta^T(t-1)\Phi(t) - \sqrt{G(t)}$, and $z_2 = \theta^T(t-1)\Phi(t) + \sqrt{G(t)}$

(ii) If $z_1 > y(t) + \gamma$, or $z_2 < y(t) - \gamma$, then the intersection of S_t and E_{t-1} is void and hence either the model or the noise bound is incorrect.

(iii) If $z_1 > y(t) - \gamma$, and $z_2 < y(t) + \gamma$, then both $H_1(t)$ and $H_2(t)$ do not intersect E_{t-1}, and hence $E_t = E_{t-1}$, is the bounding ellipsoid with minimum volume.

(iv) If $z_1 < y(t) - \gamma$, and $z_2 > y(t) + \gamma$, then both $H_1(t)$ and $H_2(t)$ intersect E_{t-1}, and hence E_t is constructed as in the MVS algorithm.

(v) If $z_1 > y(t) - \gamma$, and $z_2 > y(t) + \gamma$, then only $H_1(t)$ does not intersect E_{t-1}. Hence replace $H_1(t)$ by $H_1'(t)$, and construct the bounding ellipsoid E_t.

(vi) If $z_1 < y(t) - \gamma$, and $z_2 < y(t) + \gamma$, then only $H_2(t)$ does not intersect E_{t-1}. Hence replace $H_2(t)$ by $H_2'(t)$, and construct the bounding ellipsoid E_t.

If step (v) or (vi), the bounding ellipsoid is constructed as follows:

Define

$$f_1 = \max\,[z_1,\, y(t) - \gamma]\ ;\ f_2 = \min\,[z_2\,,\, y(t) + \gamma\,]$$

and

$$y'(t) = \frac{f_1+f_2}{2};\quad \gamma' = \frac{f_2-f_1}{2}$$

Then, it is easy to see that in step (v) or (vi), the new set S_t' is defined by

$$S_t' = \{\theta \in \mathbf{R}^N : [y'(t) - \theta^T\Phi(t)]^2 \leq \gamma'^2\ \}$$

The ellipsoid which bounds the intersection of E_{t-1} and S_t' can now be constructed exactly as in the MVS algorithm. The update equations and the equations for the optimum updating gain are exactly as in Section 2.2, except that $y(t)$ is replaced by $y'(t)$ and γ is replaced by γ'.

The modified MVS algorithm is applied to the ARX(2,2) system of Example 2.1 of the previous section. The noise sequence is white and uniformly distributed in [-1,+1]. Five Monte Carlo runs (each of 1000 data points) of the MVS and modified MVS algorithm are performed The average final volume of the bounding ellipsoid is 1.4×10^{-4}, for the modified MVS algorithm and 4×10^{-2}, for the MVS algorithm. The average final sum of semi-axes is 0.64 and 2.58 respectively. Thus the modification can cause a significant reduction in the size of the bounding ellipsoids.

An attempt was made to modify the DHOBE algorithm in the same manner. The resulting tests in the modified algorithm are as above with the only difference being that instead of using $G(t) = \Phi^T(t)\, P(t-1)\, \Phi(t)$ to construct z_1 and z_2, the quantity $G'(t) = \sigma^2(t-1)\Phi^T(t)P(t-1)\Phi(t)$ is used. When simulated, however, the modified DHOBE algorithm did not update frequently enough. This is because if γ is small for some time instant t, then $\sigma^2(t)$ tends to decrease by a large amount. Then for future time instants k, when both $H_1(t)$ and $H_2(t)$ intersect E_{t-1}, $\sigma^2(k-1)$ is much smaller than γ^2 and hence the sum $\sigma^2(k-1) + \delta^2(k)$ is much smaller than γ^2, thereby not permitting an update. Thus the modification of Belforte and Bona does not appear to be applicable to the DHOBE algorithm.

CHAPTER III

ARMA PARAMETER ESTIMATION

3.1 Introduction

In many areas of signal processing, only samples of a signal $y(t)$ are available, and it is required to obtain a model which can describe the signal as accurately as possible. For example, in speech processing, only samples of speech are available, since it is not possible to measure the glottal excitation. In seismic data processing, often only the response of the seismic layers to an excitation is measurable, while the actual probing input is unknown. In radar and array signal processing, high resolution spectral estimation is often performed by first fitting a model to the received signal.

A powerful and increasingly popular way to model the signal of interest, is to assume that it is the output of an IIR filter driven by unknown white noise $w(t)$ [Friedlander, 1982]. The signal $y(t)$ is thus modeled as an autoregressive moving average(ARMA) process of the form

$$y(t) = a_1 y(t-1)+..+ a_n y(t-n) + w(t) + c_1 w(t-1) +..+ c_r w(t-r) \qquad (3.1.1)$$

Fitting this ARMA model to the measured data $y(t)$, $t = 1,2,\ldots$ requires the estimation of the parameters $a_1 \ldots a_n$, $c_1 \ldots c_r$.

Even in cases when the input is known, as in control applications, the need for ARMA, or more accurately ARMAX parameter estimation arises. For example a DARMA model with input and output measurement noise is equivalent to an ARMAX model. This can be shown as follows. The DARMA process [Goodwin, 1984] can be expressed in the form

$$A(q^{-1})\, y(t) \;=\; B(q^{-1})\, u(t) \tag{3.1.2}$$

where $A(q^{-1}) = 1 - a_1 q^{-1} - a_2 q^{-2} \dots - a_n q^{-n}$, $B(q^{-1}) = b_0 + b_1 q^{-1} + b_2 q^{-2} + \dots + b_m q^{-m}$, and q^{-1} is the delay operator. The input $\{u(t)\}$ is assumed to be measurable. If the inputs and outputs are subject to white measurement noise, then the observed outputs $y_m(t)$ and observed inputs $u_m(t)$ are given by

$$y_m(t) = y(t) + p(t) \tag{3.1.3a}$$

$$u_m(t) = u(t) + s(t) \tag{3.1.3b}$$

where $p(t)$ and $s(t)$ are assumed to be zero mean, uncorrelated white noise processes with variances σ_p^2 and σ_s^2 respectively. Substituting (3.1.3) in (3.1.2) gives

$$A(q^{-1})\, y_m(t) \;=\; B(q^{-1})\, u_m(t) + A(q^{-1})\, p(t) - B(q^{-1})\, s(t) \tag{3.1.4}$$

By the spectral factorization theorem [Astrom, 1986], the noise terms in (3.1.4) can be replaced by a unique spectrally equivalent minimum phase process, so that (3.1.4) becomes an ARMAX process

$$A(q^{-1})\, y_m(t) \;=\; B(q^{-1})\, u_m(t) + D(q^{-1})\, w(t) \tag{3.1.5}$$

where

$$\sigma_w^2\, D(z) D(z^{-1}) = \sigma_p^2\, A(z) A(z^{-1}) - \sigma_s^2\, B(z) B(z^{-1}) \tag{3.1.6}$$

The DARMA process with input and output measurement noise is thus a special case of an ARMAX process.

Many methods for the estimation of ARMA parameters have been proposed in the literature, particularly from the spectral estimation viewpoint. Among the more recent are Cadzow's overdetermined rational equation method [Cadzow, 1982], the spectral matching technique of Friedlander and Porat [Friedlander, 1984], and the extended Yule-Walker method of Kaveh [Kaveh, 1979]. A common feature of these methods is the use of the sample autocorrelation sequence of the output process y(t). In the context of system identification, the extended least-squares (ELS), the recursive maximum likelihood (RML) and multi-stage least-squares algorithms have been used to recursively estimate ARMA parameters [Ljung, 1983, Mayne, 1982]. The ELS algorithm uses the *a posteriori* prediction error $\varepsilon(t)$, as an estimate of w(t). The regressor vector is formed from y(t-1),..., y(t-n) and $\varepsilon(t-1),\ldots,\varepsilon(t-r)$. The standard RLS algorithm is then employed to update the estimates. The algorithm is conceptually simple but restrictive in the sense that convergence of the algorithm can be assured only if the underlying transfer function $H(q^{-1}) = 1/C(q^{-1}) - 1/2$ is strictly positive real (SPR), with q^{-1} being the delay operator and

$$C(q^{-1}) = 1 + c_1 q^{-1} + c_2 q^{-2} + \ldots + c_r q^{-r} \tag{3.1.7}$$

The transfer function $H(q^{-1})$ is SPR if there exists an ε such that

$$\mathrm{Re}[H(e^{j\omega})] \geq \varepsilon > 0, \quad \text{for all } \omega \in [-\pi,\pi]$$

The RML algorithm, which uses a filtered version of the regressor vector used in the ELS algorithm, does not require $H(q^{-1})$ to be SPR. However the estimates have to be monitored and projected into a stability region to ensure convergence[Ljung, 1983].

In this chapter, the DHOBE algorithm is extended to the ARMA case. For the ARMA parameter estimation problem, since the input sequence $\{w(t)\}$ is unobservable, the DHOBE algorithm cannot be applied in its present form. However, by assuming that the input white noise is bounded in magnitude, the DHOBE algorithm can be extended in a manner similar to the ELS algorithm. The simpler optimization criterion of the DHOBE algorithm makes the convergence analysis of the extended algorithm tractable. The analysis is performed by imposing a bound on the sum of the magnitudes of the MA coefficients. This is sufficient to ensure that the true parameter vector is contained in all the optimal bounding ellipsoids. The algorithm thus gives 100% confidence regions for the parameters at every instant. The ELS or RML algorithms, in contrast can yield confidence regions only asymptotically. A uniform bound on the *a posteriori* prediction error of the extended DHOBE algorithm is derived. In contrast, even though the *a posteriori* prediction errors are generated in a stable fashion in the ELS algorithm[Ljung, 1983], it is difficult to obtain an expression for even the asymptotic bound, if such a bound exists. By imposing a persistence of excitation condition on the regressor vector, the *a priori* prediction error of the extended DHOBE algorithm is shown to be bounded and the parameter estimates are shown to converge to a neighborhood of the true parameter vector. Simulations show that the performance of the algorithm is comparable to the ELS algorithm as far as the mean-squared parameter estimation error (MSPEE) is concerned. It has also been observed , in a number of examples, that the transient performance of the extended algorithm is superior to the ELS algorithm.

3.2 Extension to ARMA Parameter Estimation

The ARMA model described by (3.1.1), can be rewritten as

$$w(t) = y(t) - \theta^{*T} \Phi'(t) \tag{3.2.1}$$

where θ^*, the vector of true parameters, and $\Phi'(t)$ are defined by

$$\theta^* = [a_1 , a_2 \ldots a_n , c_1 , c_2 \ldots, c_r]^T$$

$$\Phi'(t) = [y(t-1) , .., y(t-n) , w(t-1) , \ldots, w(t-r)]^T$$

Here again, $w(t)$ is assumed to be bounded in magnitude, i.e. there exists positive γ_0 such that

$$| w(t) | \leq \gamma_0 \tag{3.2.2}$$

Since the values of the noise sequence $\{w(t)\}$ are not available, the regressor vector $\Phi'(t)$ is not known exactly. If, however, at time t, an estimate of θ^*,

$$\theta(t) = [a_1(t), \ldots, a_n (t) \ c_1 (t), \ldots c_r (t)]^T \tag{3.2.3}$$

is available, $w(t)$ could be estimated by the *a posteriori* prediction error(also called the residual error by some authors)

$$\varepsilon(t) \;=\; y(t) - \theta^T(t)\Phi(t) \tag{3.2.4}$$

where

$$\Phi(t) = [\, y(t-1), \; .. \; y(t-n), \varepsilon(t-1), \; ... \; \varepsilon(t-r) \,]^T \tag{3.2.5}$$

Now just as in the ARX case, define for some suitable γ^2, the set

$$S_t = \{ \theta : (y(t) - \theta^T \Phi(t))^2 \leq \gamma^2, \theta \in \mathbf{R}^{n+r} \}$$

and the bounding ellipsoid

$$E_t = \{ \theta \in \mathbf{R}^{n+r} : (\theta - \theta(t))^T P^{-1}(t) (\theta - \theta(t)) \leq \sigma^2(t) \} \tag{3.2.6}$$

The update equations for $\theta(t)$, $P(t)$ and $\sigma^2(t)$ are as in Section 2.3, with the only difference being that the regressor vector is now given by (3.2.5).

As in the OBE algorithm, the bounding ellipsoids are optimized by choosing λ_t to minimize $\sigma^2(t)$. In order to facilitate the subsequent analysis, the initial conditions are modified to

$$P(0) = M\, \mathbf{I}_{n+r}, \; \theta(0) = 0, \text{ and } \; \sigma^2(0) \leq \gamma^2 \tag{3.2.7}$$

where $M \gg 1$, and $\mathbf{I}_{n+r}$ is the identity matrix of dimension n+r.

This choice of initial conditions ensures that the initial ellipsoid E_0 will contain the true parameter vector θ^* and more importantly, as shown in Appendix 3A, simplifies the optimum forgetting factor λ_t^* formula to

$$\text{If } \sigma^2(t-1) + \delta^2(t) \leq \gamma^2 \text{ then } \lambda_t^* = 0, \tag{3.2.8}$$

otherwise

$$\lambda_t^* = \begin{cases} \dfrac{1-\beta(t)}{2} & \text{if } G(t) = 1 \quad (3.2.9a) \\ \dfrac{1}{1-G(t)}\left[1-\sqrt{\dfrac{G(t)}{1+\beta(t)\,(G(t)-1)}}\right] & \text{if } G(t) \neq 1 \quad (3.2.9b) \end{cases}$$

where

$$\beta(t) \triangleq \frac{\gamma^2 - \sigma^2(t-1)}{\delta^2(t)} \quad (3.2.9c)$$

Remarks

(1) It is shown in Appendix 3A that if $\sigma^2(t-1) + \delta^2(t) > \gamma^2$ then λ_t^* given by (3.2.9) satisfies

$$\left.\frac{d\sigma^2(t)}{d\lambda_t}\right|_{\lambda_t=\lambda_t^*} = 0$$

and furthermore $0 < \lambda_t^* < 1$. Thus unlike the case in Section 2.3, no upper bound need be imposed on the forgetting factor.

(2) Since $\sigma^2(t) = \sigma^2(t-1)$ if $\lambda_t^* = 0$, any non-zero value of λ_t^* which minimizes $\sigma^2(t)$, will cause $\sigma^2(t) < \sigma^2(t-1)$. Thus choosing λ_t^* to minimize $\sigma^2(t)$, causes $\{\sigma^2(t)\}$ to be a non-increasing sequence.

The recursive relations (2.3.5) – (2.3.7), the initial conditions (3.2.7), the selective update strategy (3.2.8) and the forgetting factor determination formula (3.2.9) form the Extended Optimal Bounding Ellipsoid (EOBE) estimation algorithm. The choice of the threshold γ^2 will become clear from the analysis below. The algorithm retains the discerning update strategy and the modular adaptive filter structure of the DHOBE algorithm.

3.3 Convergence Analysis of the EOBE Algorithm

The main difficulty in the analysis of the EOBE algorithm arises from the presence of the *a posteriori* prediction errors in the regressor vector. Unlike the OBE algorithm, in this case, boundedness of w(t) does not guarantee that all the sets S_t, t=1,2,.. will contain θ^*. The first step in the analysis is to find conditions under which this happens. The minimization of $\sigma^2(t)$, at every time instant, and the choice of initial conditions (3.2.7), facilitate the characterization of the behavior of the *a posteriori* prediction errors.

Lemma 3.1. For the EOBE algorithm, if $\sigma^2(t-1) + \delta^2(t) > \gamma^2$, i.e., if an update occurs at time instant t, then

(i) $\sigma^2(t) + \varepsilon^2(t) = \gamma^2$, (3.3.1)

(ii) $\varepsilon^2(k) \leq \varepsilon^2(t)$ for all time instants $k < t$, (3.3.2)

and if t+j is the time instant at which the next update occurs then

(iii) $\varepsilon^2(k) \leq \varepsilon^2(t)$ for all $k < t+j$. (3.3.3)

Proof:

(i) It has been shown in Appendix 3A, that if $\sigma^2(t-1) + \delta^2(t) > \gamma^2$ then, the optimum forgetting factor λ^*_t satisfies

$$\left.\frac{d\,\sigma^2(t)}{d\,\lambda_t}\right|_{\lambda_t = \lambda_t^*} = 0 \tag{3.3.4}$$

Taking the derivative in (2.3.5) and using (3.3.4) yields

$$\gamma^2 - \sigma^2(t-1) - \frac{(1-\lambda_t)\,\delta^2(t)}{1-\lambda_t+\lambda_t G(t)} = -\frac{\lambda_t \delta^2(t)\, G(t)}{(1-\lambda_t+\lambda_t G(t))^2} \tag{3.3.5a}$$

which can be rewritten in the form

$$\gamma^2 - \sigma^2(t-1) = \delta^2(t)\,\frac{(1-\lambda_t)^2 - \lambda_t^2 G(t)}{(1-\lambda_t+\lambda_t G(t))^2} \tag{3.3.5b}$$

In (3.3.5) and in the remainder of the chapter, when there is no risk of confusion, the optimum forgetting factor λ_t^* will be denoted by λ_t. The *a posteriori* prediction error

$$\varepsilon(t) = y(t) - \theta^T(t)\Phi(t) = y(t) - \theta^T(t-1)\Phi(t) - \lambda_t \Phi^T(t) P(t)\Phi(t)\,\delta(t)$$

Thus

$$\varepsilon(t) = [1 - \lambda_t \Phi^T(t) P(t)\Phi(t)]\,\delta(t)$$

Multiplying P(t) in (2.3.7) on the left by $\Phi^T(t)$ and right by $\Phi(t)$ yields

$$1-\lambda_t\Phi^T(t)P(t)\Phi(t) = \frac{1-\lambda_t}{1-\lambda_t+\lambda_t G(t)}$$

Thus

$$\varepsilon(t) \quad = \quad \frac{1-\lambda_t}{1-\lambda_t+\lambda_t G(t)}\,\delta(t) \tag{3.3.6}$$

Note that the non-negativeness of G(t) implies that $\varepsilon^2(t) \le \delta^2(t)$. Substituting (3.3.6) in (3.3.5b) and rearranging terms yields

$$(1-\lambda_t)\,\gamma^2 - (1-\lambda_t)\,\sigma^2(t-1) = (1-\lambda_t)\,\varepsilon^2(t) - \frac{\lambda_t^2 G(t)\,\varepsilon^2(t)}{1-\lambda_t} \tag{3.3.7}$$

Now using (3.3.6) in (2.3.5) gives

$$\sigma^2(t) \quad = \quad (1-\lambda_t)\,\sigma^2(t-1) + \lambda_t\gamma^2 - \lambda_t\varepsilon^2(t) - \frac{\lambda_t^2 G(t)}{1-\lambda_t}\,\varepsilon^2(t) \tag{3.3.8}$$

Finally, subtracting (3.3.8) from (3.3.7) gives (3.3.1).

(ii) <u>Case 1</u> If k < t, is an updating instant. Then (3.3.1) gives

$$\sigma^2(k) + \varepsilon^2(k) \;=\; \gamma^2 \tag{3.3.9}$$

But since $\{\sigma^2(t)\}$ is a non-increasing sequence, (3.3.9) and (3.3.1) together would imply that

$$\varepsilon^2(k) \;\le\; \varepsilon^2(t)$$

<u>Case 2</u> If k < t, is a non-updating instant, then $\varepsilon^2(k) = \delta^2(k)$ and so by (2.3.8),

$\sigma^2(k-1) + \varepsilon^2(k) \le \gamma^2$, and since $\sigma^2(t)$ is non-increasing, $\varepsilon^2(k) \le \varepsilon^2(t)$.

(iii) Since λ_k, $k = t+1, t+2,\ldots, t+j-1$, are all zero, $\sigma^2(k) = \sigma^2(t)$, for all $t < k < t+j$. And because k is a non-updating instant, $\sigma^2(k-1) + \varepsilon^2(k) = \sigma^2(t) + \varepsilon^2(k) \le \gamma^2$, and so (3.3.3) follows.

Remark

Lemma 3.1 shows that the the sequence of squared residual errors evaluated at the updating instants is non-decreasing. Furthermore, the squared residual error at any non-updating instant is not greater than the squared residual errors at the updating instants immediately prior and after the non-updating instant. This characterization of the squared residual errors is useful in deriving sufficient conditions under which the convex polytopes S_t and E_t will contain θ^*.

Theorem 3.1. The sets S_t and consequently the ellipsoids E_t, $t = 1,2,\ldots$ will contain the true parameter, if

(i) E_0 contains θ^*. (3.3.10a)

(ii) The true moving average coefficients satisfy

$$\sum_{i=1}^{r} |c_i| < 1.0 \tag{3.3.10b}$$

(iii)The threshold γ satisfies

$$\gamma \ge \frac{\gamma_0\left(1+\sum_{i=1}^{r}|c_i|\right)}{1-\sum_{i=1}^{r}|c_i|} \tag{3.3.10c}$$

Proof. Let the induction hypothesis be $\theta^* \in E_{t-1}$. Then defining

$$V(t) = (\theta(t) - \theta^*)^T P^{-1}(t) (\theta(t) - \theta^*) \tag{3.3.11}$$

and recalling the definition of E_{t-1} yields

$$V(t-1) \leq \sigma^2(t-1) \tag{3.3.12}$$

and since $P^{-1}(t)$ is positive definite for all t. $\sigma^2(t-1) \geq 0$. Now using (3.2.1) and (3.2.5) yields

$$y(t) - \theta^{*T}\Phi(t) = C(q^{-1})[w(t)] - (C(q^{-1}) - 1)[\varepsilon(t)] \tag{3.3.13}$$

where the operator $C(q^{-1})$ has been defined in (3.1.7). Defining $n(t) = C(q^{-1})[w(t)]$ then yields

$$|y(t) - \theta^{*T}\Phi(t)| \leq |n(t)| + |c_1\varepsilon(t-1) + c_2\varepsilon(t-2) ... + c_r\varepsilon(t-r)|$$

But

$$|n(t)| \leq \gamma', \text{ for all } t$$

where

$$\gamma' \triangleq \gamma_0 (1 + \sum_{i=1}^{r} |c_i|) \tag{3.3.14}$$

Hence

$$|y(t) - \theta^{*T}\Phi(t)| \leq \gamma' + (|c_1||\varepsilon(t-1)| + |c_2||\varepsilon(t-2)| + ... + |c_r||\varepsilon(t-r)|) \tag{3.3.15}$$

But by Lemma 3.1. if t-j is the updating instant immediately preceding time instant t. then

$$|\varepsilon(t-i)| \leq |\varepsilon(t-j)| \text{ for } 1 \leq i \leq r$$

Thus

$$|y(t) - \theta^{*T}\Phi(t)| \leq \gamma' + |\varepsilon(t-j)| \sum_{i=1}^{r} |c_i| \tag{3.3.16}$$

Now $\varepsilon^2(t-j) = \gamma^2 - \sigma^2(t-j) = \gamma^2 - \sigma^2(t-1)$. By the induction hypothesis. $\sigma^2(t-1) \geq 0$. Hence

$$|\varepsilon(t-j)| \leq \gamma$$

and so

$$|y(t) - \theta^{*T}\Phi(t)| \leq \gamma' + \gamma \sum_{i=1}^{r} |c_i|$$

So S_t will contain θ^* if

$$\gamma' + \gamma \sum_{i=1}^{r} |c_i| \leq \gamma \tag{3.3.17}$$

The inequality (3.3.17) will hold iff (3.3.10b) and (3.3.10c) are true. Assuming (3.3.10b) and (3.3.10c) thus guarantees that for all time instants t

$$(y(t) - \theta^{*T}\Phi(t))^2 \leq \gamma^2 \tag{3.3.18}$$

Using (2.3.6) and (2.3.5), it is easy to show that

$$V(t) = (1-\lambda_t)V(t-1) + \lambda_t\left(y(t) - \theta^{*T}\Phi(t)\right)^2 - \frac{\lambda_t(1-\lambda_t)\delta^2(t)}{1-\lambda_t+\lambda_t G(t)} \tag{3.3.19}$$

Now using (2.3.5) in the above equation yields

$$V(t) - \sigma^2(t) \leq (1-\lambda_t)(V(t-1) - \sigma^2(t-1)) + \lambda_t[(y(t) - \theta^{*T}\Phi(t))^2 - \gamma^2]$$

and so from (3.3.18) it follows that

$$V(t) - \sigma^2(t) \leq (1-\lambda_t)(V(t-1) - \sigma^2(t-1)) \tag{3.3.20}$$

Finally, by (3.3.12), it follows that

$$V(t) - \sigma^2(t) \leq 0 \tag{3.3.21}$$

i.e., E_t contains θ^*, and $\sigma^2(t)$ is non-negative for all t.

Remarks

(1) The condition (3.3.10b) implies that the noise sequence $n(t) = C(q^{-1})[w(t)]$ should not be " too colored ". It is interesting to see how this this condition relates to similar conditions which appear in the convergence analysis of other parameter estimation algorithms. A sufficient condition for convergence of the ELS algorithm is that the transfer function $[1/C(q^{-1}) - 1/2]$ be SPR. It is shown in Appendix 3B that, for this transfer function to be SPR, it is necessary that

$$\sum_{i=1}^{r} |c_i|^2 < 1 \tag{3.3.22}$$

Thus if (3.3.22) is not satisfied, the transfer function is not SPR and so the ELS algorithm cannot be guaranteed to converge. Coincidentally enough, the condition (3.3.10b) is identical to the Strictly Dominant Passive (SDP) condition [Dasgupta, 1987] which appears in the analysis of some signed LMS algorithms. In Appendix 3B, it is also shown that if (3.3.10b) holds then $[1/C(q^{-1}) - 1/2]$ is SPR. However, the converse is not true. For example if $C(q^{-1}) = 1 + 0.8\, q^{-1} + 0.15\, q^{-1} - 0.15\, q^{-2}$ then $[1/C(q^{-1}) - 1/2]$ is SPR, but $C(q^{-1})$ does not satisfy (3.3.10b). Thus this condition is more restrictive than the SPR condition.

(2) Selection of the right " noise bound" γ^2 is made possible by (3.3.10c). The user would, however, need to have some knowledge of the magnitude of the true moving average coefficients. It is interesting to see that as the moving average filtering of $\{w(t)\}$ increases ($\Sigma|c_i|$ increases), the bound γ^2 is required to increase, to guarantee that the true parameter is contained in all the sets S_t and ellipsoids E_t. Simulation results show that overestimation of γ^2 has very little effect on the parameter estimates (centers of the bounding ellipsoids), though it may have an adverse effect on the size of the bounding ellipsoids.

(3) The theorem shows that $V(t) \le \sigma^2(t)$ for all t, Since $\sigma^2(0) \le \gamma^2$, and $\{\sigma^2(t)\}$is non-increasing, the theorem provides instantaneous bounds on the normalized parameter estimation error V(t).

(4) The conditions (3.3.10b,3.3.10c) are not necessary conditions and the algorithm has been observed to perform well in several examples where these conditions were violated.

The following result follows straightforwardly from Lemma 3.1 and Theorem 3.1.

Corollary 3.1. If the conditions of Theorem 3.1 hold then

(a) $$\lim_{t_j \to \infty} \varepsilon^2(t_j) \quad \text{exists} \tag{3.3.23a}$$

where $\{t_j\}$ is the subsequence of updating instants of the EOBE algorithm, and

(b) Uniformly bounded *a posteriori* prediction errors

$$\varepsilon^2(t) \le \gamma^2, \text{ for all time instants } t \tag{3.3.23b}$$

Boundedness of $\delta^2(t)$, the *a priori* prediction error, and convergence of the parameter estimates to a neighborhood of the true parameter can be assured, by requiring the regressor vector to be persistently exciting. The next lemma relates the positive definiteness of $P^{-1}(t)$ to the richness of the regressor vector $\Phi(t)$.

Lemma 3.2. If there exist positive α_3 and N such that, for all t

$$\sum_{i=t}^{t+N} \Phi(i)\,\Phi^T(i) \;\ge\; \alpha_3 I \;>\; 0 \tag{3.3.24a}$$

then there exists a positive α_4 such that

$$P^{-1}(t) \ge \alpha_4 I > 0 \tag{3.3.24b}$$

Proof of the lemma is the same as that of Theorem 4.1 of [Dasgupta, 1987], it is thus omitted here.

Remark. The positive definiteness of $P^{-1}(t)$ implies that the eigenvalues of P(t) are upper bounded.

Theorem 3.2. If the assumptions of Theorem 3.1 are satisfied and (3.3.24a) holds then the EOBE algorithm ensures :

(a) Parameter difference convergence

$$\lim_{t \to \infty} \| \theta(t) - \theta(t-k) \| = 0 \tag{3.3.25}$$

for any finite k.

(b) If, in addition, the process (1.1) is stable then the algorithm yields asymptotically bounded *a priori* prediction errors

$$\delta^2(t) \rightarrow [\,0\,,\gamma^2\,] \tag{3.3.26}$$

(c) If the moving average coefficients are further restricted to satisfy

$$[\sum_{i=1}^{r} |\,c_i\,|\,]^2 < 0.5$$

then an asymptotic bound on the parameter estimation error can be obtained

$$\|\theta(t)-\theta^*\|^2 \rightarrow [\,0,\;2\,\gamma_0^2\,(1+\Sigma\,|c_i\,|\,)^2 / \alpha_4\,] \tag{3.3.27}$$

where γ_0^2 and α_4 are as in (3.2.2) and (3.3.24b) respectively.

Proof.

(a) From (2.3.6) and (2.3.7)

$$\|\,\theta(t) - \theta(t\text{-}1)\,\|^2 = \frac{\lambda_t^2\;\Phi^T(t)\,P^2(t\text{-}1)\,\Phi(t)\,\delta^2(t)}{(\,1\text{-}\,\lambda_t + \lambda_t\,G(t)\,)^2} \tag{3.3.28}$$

$$\leq e_{max}\{\,P(t\text{-}1)\,\}\,\frac{\lambda_t^2\;\;G(t)\,\delta^2\,(t)}{(1\,\text{-}\,\lambda_t + \lambda_t\,G(t)\,)^2} \tag{3.3.29}$$

where $e_{max}\{P(t\text{-}1)\}$ is the maximum eigenvalue of P(t-1), and $\|.\|$ denotes the euclidean norm. To see how (3.3.29) follows from (3.3.28), consider any positive semi-definite symmetric matrix A and vector x. Then

$$x^T A x = x^T A\,A^{-1}\,A x = y^T A y$$

where $y = Ax$. But

$$y^T A y \geq e_{min}[A^{-1}]y^T y$$

where e_{min} refers to the minimum eigenvalue. Since $e_{max}(A) = 1/\,e_{min}(A^{-1})$, hence

$$y^T y \leq e_{max}[A]x^T A x$$

and thus

$$x^T A^2 x \leq e_{max}[A]x^T A x$$

Now multiplying both sides of (3.3.5a) by λ_t, and using (2.3.5) yields

$$\sigma^2(t) = \sigma^2(t-1) - \frac{\lambda_t^2 \ \delta^2(t)\, G(t)}{(1-\lambda_t + \lambda_t\, G(t))^2} \qquad (3.3.30)$$

The non-negativity of $\sigma^2(t)$ therefore implies

$$\sum_{i=1}^{t} \frac{\lambda_i^2 \ \delta^2(i)\, G(i)}{(1-\lambda_i + \lambda_i\, G(i))^2} = \sigma^2(0) - \sigma^2(t) < \infty \qquad (3.3.31)$$

Hence

$$\lim_{t \to \infty} \frac{\lambda_t^2 \ \delta^2(t)\, G(t)}{(1-\lambda_t + \lambda_t\, G(t))^2} = 0 \qquad (3.3.32)$$

If (3.3.24a) holds then by Lemma 3.2, $e_{max}\{P(t-1)$, the maximum eigenvalue of P(t-1), is bounded for all t, and hence (3.3.29) and (3.3.32) yield

$$\| \theta(t) - \theta(t-1) \| \to 0 \qquad (3.3.33)$$

Applying the Minkowski inequality to $\|\theta(t)-\theta(t-k)\|$ and using (3.3.33) completes the proof of (3.3.25).

(b) Stability of the process (3.1.1) and the boundedness of w(t) implies that the outputs y(t) are bounded. Hence from (2.3.16) (3.3.23b), and Lemma 3.2, it follows that

$$G(t) \le e_{max}\{P(t-1)\}[\, r\gamma^2 + n \max_{t-n \le i \le t-1} y^2(i)\,] < \infty$$

where n is the order of the AR process and r is the order of the MA process. It can now be shown, just as in Theorem 3.2 of [Dasgupta, 1987], that the *a priori* prediction errors satisfy (3.3.26).

(c) From (3.3.16)

$$\left(y(t) - \theta^{*T}\Phi(t)\right)^2 \le 2\gamma'^2 + 2\left[\sum_{i=1}^{r} |c_i|\right]^2 \varepsilon^2(t-j)$$

Assuming

$$\left[\sum_{i=1}^{r} | c_i | \right]^2 < 0.5$$

then yields

$$\left(y(t) - \theta^{*T} \Phi(t)\right)^2 \leq 2\gamma'^2 + \varepsilon^2(t - j)$$

Substituting in (3.3.19) and using (3.3.6) gives

$$V(t) = (1 - \lambda_t)V(t - 1) + \lambda_t \left[2\gamma'^2 + \varepsilon^2(t - j) - \varepsilon^2(t) - \frac{\lambda_t G(t)}{1 - \lambda_t} \varepsilon^2(t) \right] \qquad (3.3.34)$$

From Lemma 3.1, $\varepsilon^2(t\text{-}j) \leq \varepsilon^2(t)$, where t-j is the updating instant prior to time instant t.

Thus

$$V(t) - 2\gamma'^2 \leq (1 - \lambda_t) [V(t\text{-}1) - 2\gamma'^2]$$

For large enough t, the term on the right hand side goes to zero and hence for large enough t

$$V(t) = (\theta(t) - \theta^*)^T P^{-1}(t) (\theta(t) - \theta^*) \leq 2\gamma'^2$$

And so (3.3.27) follows from Lemma 3.2 and (3.3.14).

Remarks:

(1) The results of Theorem 3.1, and the results (3.3.25), (3.3.26) of Theorem 3.2, do not require the process to be stable ($A(q^{-1}) = 1 - a_1 q^{-1} - a_2 q^{-2} - \ldots - a_n q^{-n}$, to be Hurwitz). However if the process is unstable, then the *a priori* prediction errors will become very large, thereby causing overflows. In addition, on account of finite precision effects, the matrix P(t) may not stay positive definite, thus invalidating the notion of bounding ellipsoids and causing the algorithm to fail. In this situation, the ELS algorithm will fail, too.

(2) Theorems 3.1 and 3.2, do not impose any statistical properties on the input sequence$\{w(t)\}$. However our simulation experience has been that the parameter

estimates are biased if the input is not white. Of course, such is also the case, for the ELS algorithm. The EOBE algorithm uses the boundedness property of the inputs to construct confidence regions (ellipsoids) for the parameters, irrespective of the color of the inputs. This feature is absent in any of the existing ARMA parameter estimation algorithms.

(3) The upper bound given by (3.3.27) is usually looser than the bound $\| \theta(t) - \theta^* \|^2 \leq \sigma^2(t) / \alpha_4$, yielded by (3.3.21).

3.4 Simulation Studies

Simulations have been performed to investigate the performance of the EOBE algorithm *vis a vis* the ELS algorithm. In this paper, we present simulation results for three examples- a broad band ARMA (3,3) process, a narrow band ARMA(2,2) process and an ARMAX(3,3,2) process.

Example 3.1 Broad band ARMA (3,3) process

The output data $\{y(t)\}$ is generated by the following difference equation

$$y(t) = -0.4\, y(t-1) + 0.2\, y(t-2) + 0.6\, y(t-3) + w(t) - 0.22\, w(t-1) + 0.17\, w(t-2) - 0.1\, w(t-3)$$

The noise sequence $\{w(t)\}$ is generated by a pseudo-random number generator with a uniform probability distribution in [-1,1]. The upper bound γ^2 was set equal to 25.0, and $\sigma^2(0) = \gamma^2 - 0.1$. The parameter estimates were obtained by applying the EOBE algorithm to 1000 point data sequences. Twenty five runs of the algorithm were performed on the same model but with different input noise sequences. The average squared parameter error $L_1(t)$, is computed for the AR coefficients according to the formula

$$L_1(t) = \frac{1}{25} \sum_{j=1}^{25} l_j(t)$$

where $l_j(t)$, the squared AR parameter error at time t for the j'th run, is defined by

$$l_j(t) = \sum_{i=1}^{n} (a_i(t) - a_i)^2$$

with a_i and $a_i(t)$ being defined by (1.1) and (3.3), respectively. The average squared parameter error $L_2(t)$ for the MA coefficients is defined analogously. Fig. 3.1 and Fig. 3.2 display the average squared estimation errors for the AR and MA parameters using both the EOBE and the ELS algorithms. The curves show that the performance of the two algorithms is comparable. The average number of updates for the EOBE algorithm was 160 for 1000 point data sequences. Thus only 16% of the samples are used for updates, as compared to the ELS algorithm which updates at every sampling instant.

The effect of different choices for the upper bound γ^2 on the performance has also been studied. For each value of γ^2, the asymptotic mean-squared parameter estimation error (MSPEE), was computed over 25 runs of the algorithm, according to the formula

$$\text{MSPEE} = \frac{1}{25} \sum_{j=1}^{25} \| \theta_j(1000) - \theta^* \|^2$$

where $\theta_j(1000)$ is the parameter estimate at the 1000'th iteration, in the j'th run. The lower bound on γ^2 as calculated from (3.3.10c) is $\gamma^2 \geq 8.54$. The second column of Table 3.1 lists the different values of MSPEE obtained when γ^2 is varied from 0.5 to 100. In each case, $\sigma^2(0) = \gamma^2 - 0.1$. The third column lists the average number of updates. The fourth and fifth columns enumerate the average number of times the true parameter is not contained in S_t and E_t respectively. The last two columns provide measures of the average final volume and average final sum of semi-axes respectively. It is clear that the centers of the bounding ellipsoids are insensitive to the value of γ^2, since the tap error is almost constant, though the final size of the ellipsoids is very sensitive to γ^2. This can be explained as follows.

The update equations (2.3.6) and (2.3.4) for the parameter estimates and the matrix

P^{-1} , can be affected by γ^2 only through the forgetting factor λ_t. From (3.2.8), it is clear that the update decision depends on whether $\delta^2(t)$ is greater than or less than $[\gamma^2 - \sigma^2(t-1)]$. If $\delta^2(t)$ is greater than $[\gamma^2 - \sigma^2(t-1)]$ then, from (3.2.9), the calculated value of λ_t again depends on γ^2 only through $[\gamma^2 - \sigma^2(t-1)]$. But from (2.3.5), $\gamma^2 - \sigma^2(t)$ depends on γ^2 only through the quantities $\gamma^2 - \sigma^2(t-1)$, $\gamma^2 - \sigma^2(t-2)$, ..., $\gamma^2 - \sigma^2(0)$. So for all time instants t, $\gamma^2 - \sigma^2(t)$ is a complicated function of the data, $\theta(0)$, $P(0)$, and $[\gamma^2 - \sigma^2(0)]$, and consequently, λ_t, $P(t)$ and $\theta(t)$ depend on γ^2 only through $[\gamma^2 - \sigma^2(0)]$. Thus, since this difference is constant in the above simulations, the parameter estimates are unaffected by the value of γ^2. However, the final size of the ellipsoids depends on $\sigma^2(t)$, and since $\gamma^2 - \sigma^2(1000)$ is constant for all values of γ^2, the final size is proportional to γ^2. When $\gamma^2 = 0.5$, $\sigma^2(1000)$ is negative and hence the final ellipsoids do not exist.

TABLE 3.1

γ^2	T (dB)	Avg. updts	Avg. $\theta^* \notin S_t$	Avg. $\theta^* \notin E_t$	Avg. final volume	Avg. final sum of semi-axes
0.5	-15.34	156	292	949	-	-
1.0	-15.34	156	13	0	0.22	10.46
2.0	-15.36	156	0	0	2.6×10^4	74.08
5.0	-15.34	156	0	0	5.4×10^7	264.91
25.0	-15.35	156	0	0	2.1×10^{12}	1536.92
100.0	-15.39	156	0	0	1.0×10^{16}	6303.29

The performance of the algorithm, when the noise sequence $\{w(t)\}$ has a gaussian distribution, was evaluated in a similar fashion. A constant value of $\gamma^2 = 25$ was used and the standard deviation of the input was varied. The results for 25 runs of the algorithm are shown in Table 3.2. It is clear that the unbounded noise has marginal effect on the parameter estimates.

TABLE 3.2

S.D of input	T (dB)	Avg. updts	Avg. $\theta^* \notin S_t$	Avg. $\theta^* \notin E_t$	Avg. final volume	Avg. final sum of semi-axes
0.5	-4.9	90	0	0	5.6×10^{12}	1574.6
1.0	-5.95	105	0	0	4.89×10^{8}	333.4
2.0	-5.98	114	12	26	1.47×10^{2}	19.5
5.0	-6.29	119	323	965	-	-

The tracking capability of the EOBE algorithm was compared with that of the ELS algorithm (with forgetting factor=0.99). The same model was used to generate 400 data points. The parameters were then changed by 150% and the next 400 points were generated. Finally the last 200 points were generated by using the original parameters. The average squared parameter error was evaluated over 25 runs and is shown in Figure 3.3. Even though the formulation of bounding ellipsoids is based on the assumption that the parameters are constant, the simulation results show that the algorithm is able to accommodate changes in model parameters. Analysis of the tracking ability of the algorithm is the focus of Chapter V.

As mentioned earlier, the transient behavior of the EOBE algorithm has been observed to be superior to that of the ELS algorithm in a number of examples. To demonstrate this, ten Monte Carlo runs of the EOBE and ELS algorithms are performed with data records of 50 points each.The parameter estimation error at each instant, $(\theta^*-\theta(t))^T(\theta^*-\theta(t))$ and the *a priori* prediction error are averaged over the ten runs and displayed in Fig. 3.4. and Fig. 3.5 respectively. The parameter estimates of the ELS algorithm tend to wander outside the stability region in the transient stage, thus causing unacceptably high prediction error bursts. The inherent stability mechanism of the ELS algorithm, however, ensures that the estimates do return to the stability region. The transient estimation error of the EOBE algorithm, in contrast, is well behaved.

Example 3.2 Narrow band ARMA (2,2) process

The output data $\{y(t)\}$ is generated by the following difference equation

$$y(t) = 1.4\, y(t-1) - 0.95\, y(t-2) + w(t) - 0.86\, w(t-1) + 0.431\, w(t-2)$$

Note that in this case, condition (3.3.10 b) of Theorem 3.1 is violated. The noise sequence is uniformly distributed in [-1.0,1.0], as in the first example. The upper bound γ^2 was set equal to 25.0. The average squared AR and MA parameter estimation errors are calculated over twenty five runs and plotted in Fig. 3.6 and Fig. 3.7 respectively. The average number of updates was 78 for 1000 point data sequences.

For this example too, different values of the upper bound γ^2 were used and no significant difference in the quality of estimates, number of updates or convergence rate was observed. Thus, it is verified once again that a precise knowledge of the upper bound is not a prerequisite for satisfactory performance of the algorithm.

Example 3.3 ARMAX(3,3,2) process

$$y(t) = 0.6y(t-1) - 0.3y(t-2) + 0.25y(t-3) + 3.8u(t) - 1.8u(t-1) + 0.7u(t-2) + w(t) + 0.4w(t-1) - 0.1w(t-2)$$

The measurable input sequence $\{u(t)\}$and the non-measurable input/noise sequence $\{w(t)\}$ are uncorrelated white noise sequences uniformly distributed in [-1,1]. The EOBE algorithm can be used for ARMAX parameter estimation, by simply increasing the dimension of the parameter vector and extending the regressor vector to include the observable inputs. The analysis performed in the previous section is still valid. In the ARMAX estimation problem, if estimates of the MA coefficients (the coefficients of $C(q^{-1})$), are not required, then the OBE algorithm can also be used, by modeling the system with an ARX model. For the EOBE algorithm, $\gamma^2 = 10$ and $\sigma^2(0)=10$. For the OBE algorithm, $\gamma^2 = 2.5$, and $\sigma^2(0)=10$. $P(0) = I$, in both cases. Ten Monte Carlo runs of 1000 data points each were performed for the EOBE and OBE algorithms.The average final parameter estimation error for the EOBE algorithm was 0.157 and the corresponding error for the OBE algorithm was 0.69. For the EOBE algorithm

(i) The sample mean

$$E[\theta(1000)] = [0.53, -0.27, 0.24, 3.79, -1.52, 0.63, 0.45, -0.07]^T$$

(ii) The average final volume = 6.2×10^{10}

(iii) The average final sum of semi-axes = 498.47

(iv) The average number of updates = 61

For the OBE algorithm

(i) $E[\theta(1000)] = [0.76, -0.42, 0.27, 3.82, -2.41, 1.11]^T$

(ii) The average final volume = 224.25

(iii) The average final sum of semi-axes = 112.47

(iv) The average number of updates = 128

Thus the parameter estimates of the OBE algorithm are biased. For the OBE algorithm, the noise $v(t) = C(q^{-1})[w(t)]$, and the correlation in $v(t)$ is responsible for the bias. The EOBE algorithm in contrast, yields unbiased estimates, however the confidence regions for the parameters are larger than the confidence regions yielded by the OBE algorithm.

3.5 Concluding Remarks

The distinctive features of the EOBE algorithm are (i) the discerning update strategy, (ii) the uniform boundedness of the *a posteriori* errors and, (iii) the fact that the true parameters are guaranteed to lie in the bounding ellipsoids at every time instant. Unfortunately, the size of these bounding ellipsoids is very sensitive to the choice of threshold γ^2. Simulations show that the threshold can be taken much smaller than the theoretical minimum, thereby obtaining tighter confidence regions, without causing the true parameter vector to slip out of any of the sets S_t. In an effort to obtain smaller ellipsoids, the MVS algorithm was extended, just as in Section 3.2. Unfortunately on account of the more complex optimization criterion used in the MVS algorithm, the counterparts of Theorem 3.1 and Theorem 3.2 could not be derived. Simulation studies indicate that the extended MVS algorithm does yield smaller bounding ellipsoids, however, the parameter estimates are biased. It is conjectured that the observed unbiasedness of the EOBE parameter estimates is due to the optimization criterion which is an upper bound on the normalized parameter estimation error.

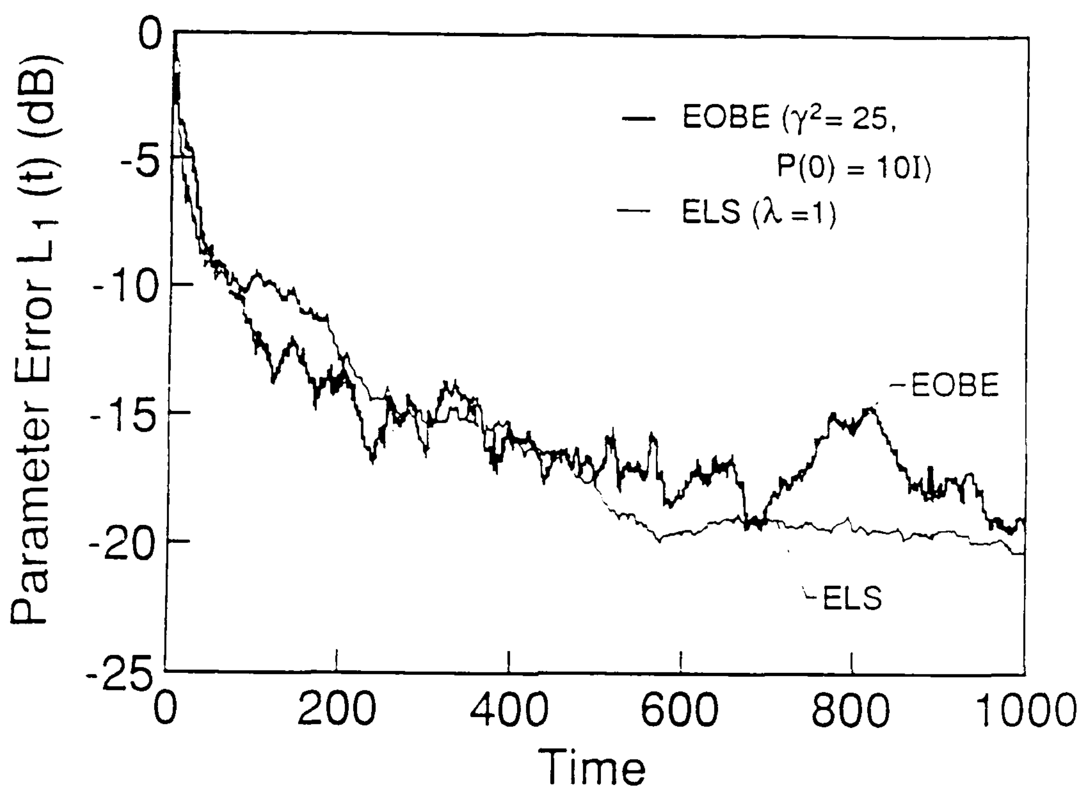

Figure 3.1 Mean-squared AR parameter estimation error - Example 3.1

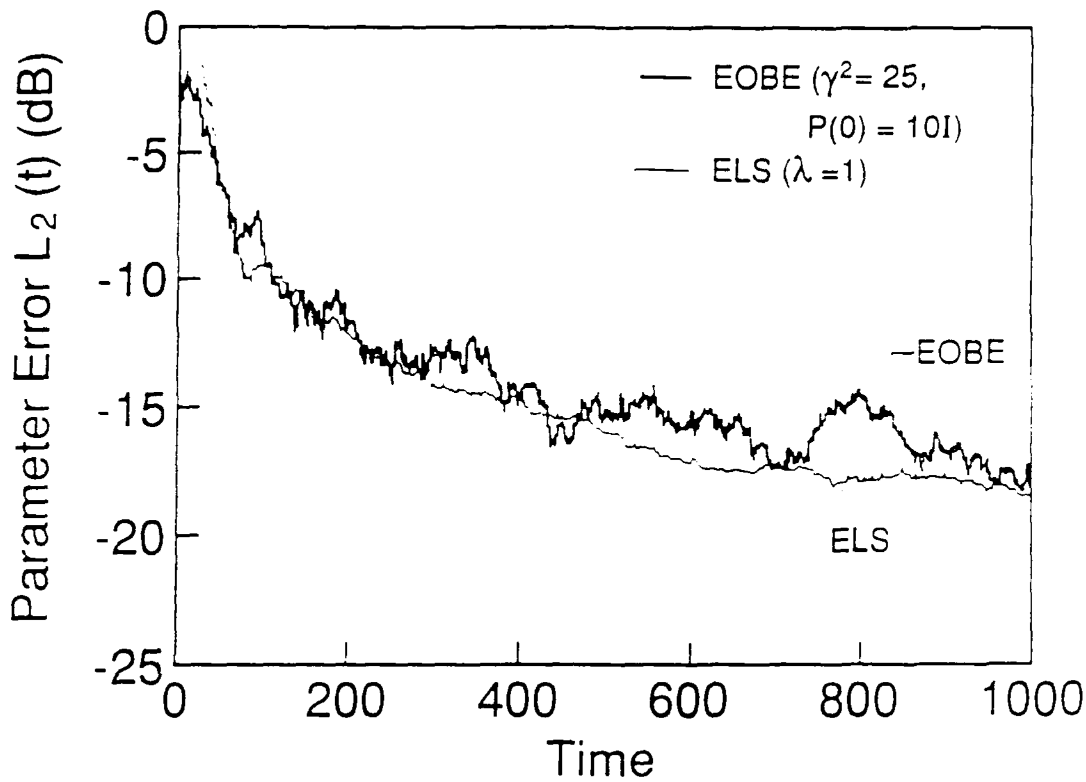

Figure 3.2 Mean-squared MA parameter estimation error - Example 3.1

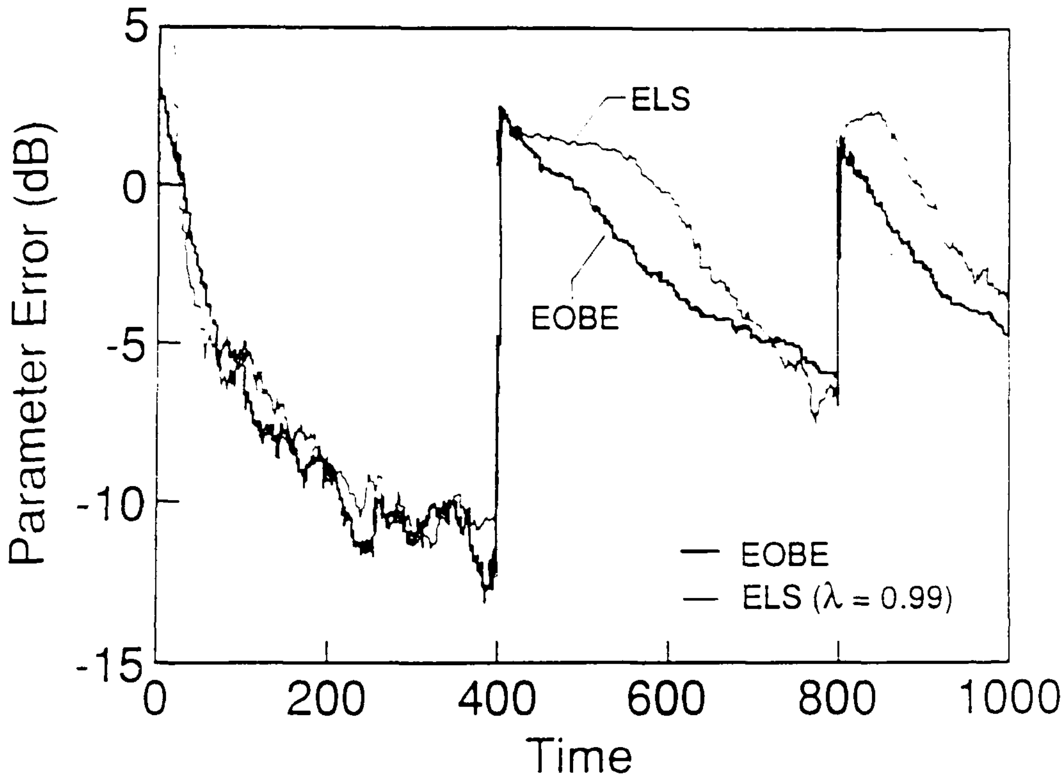

Figure 3.3 Mean-squared parameter estimation error - time varying case

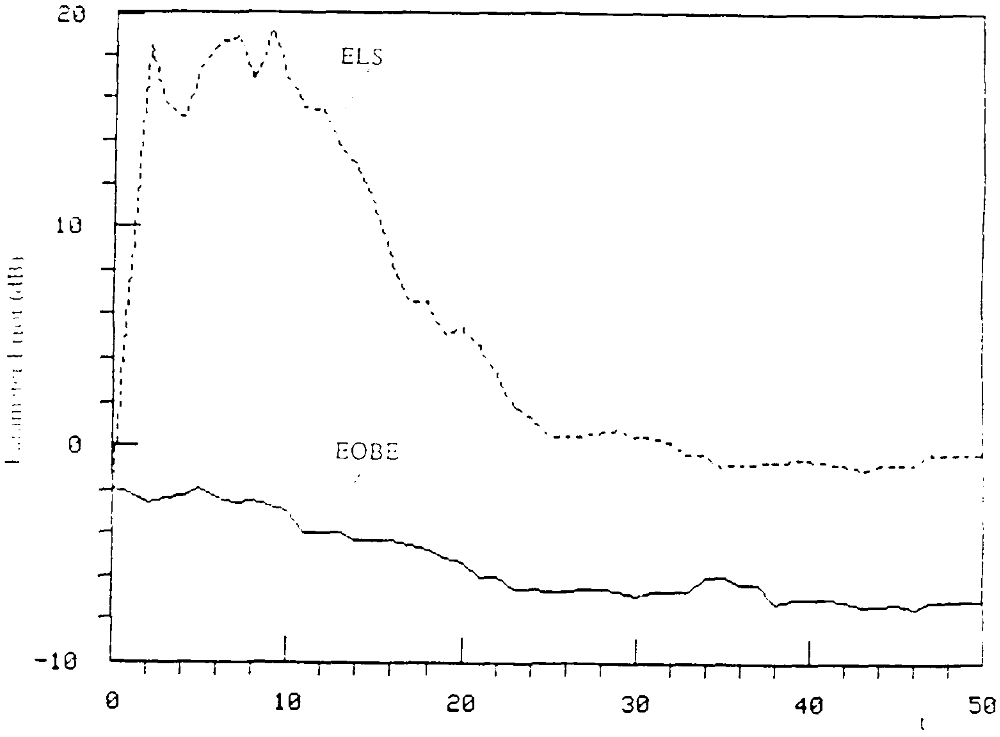

Figure 3.4 Mean-squared parameter estimation error in the transient stage

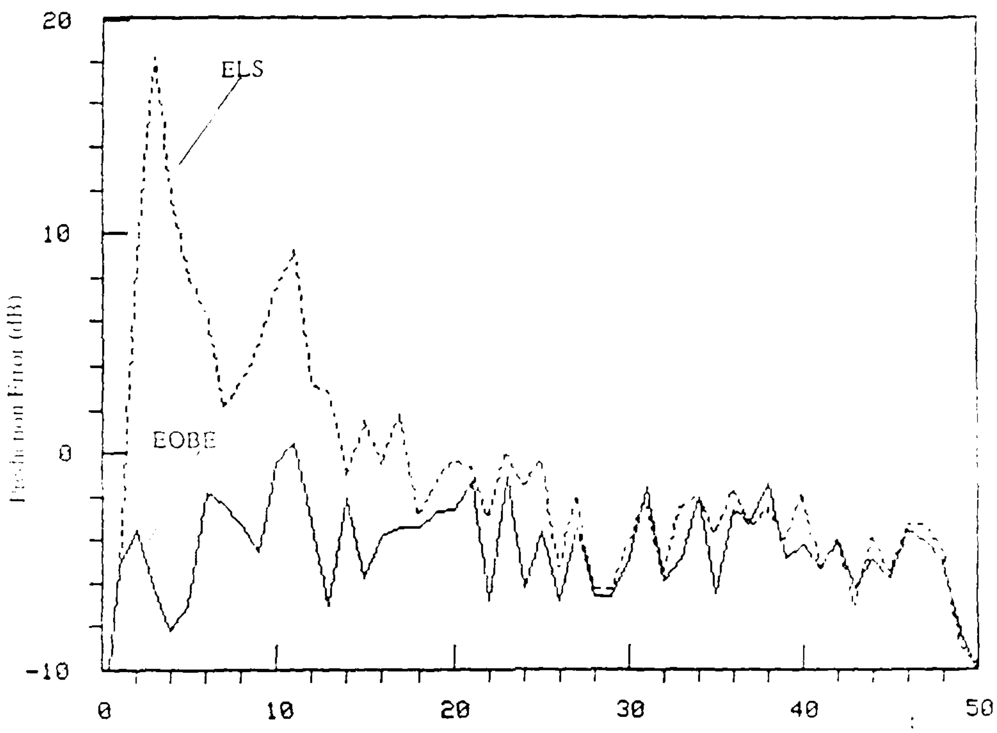

Figure 3.5 Mean-squared prediction error in the transient stage

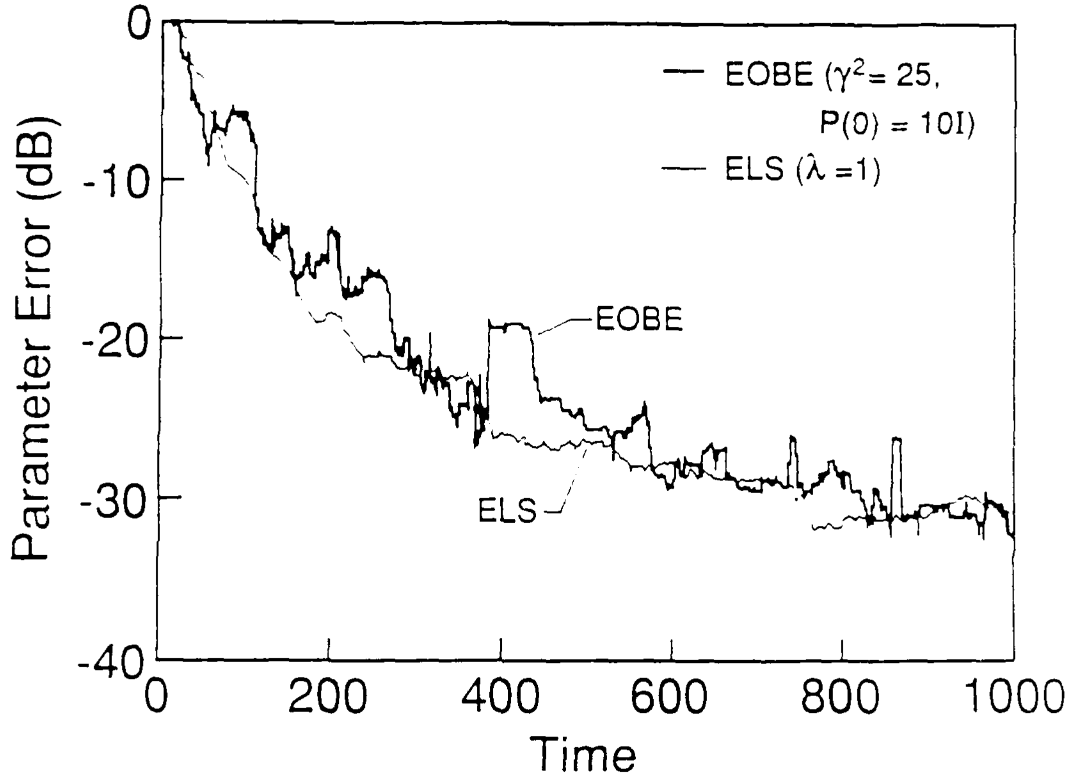

Figure 3.6 Mean-squared AR parameter estimation error - Example 3.2

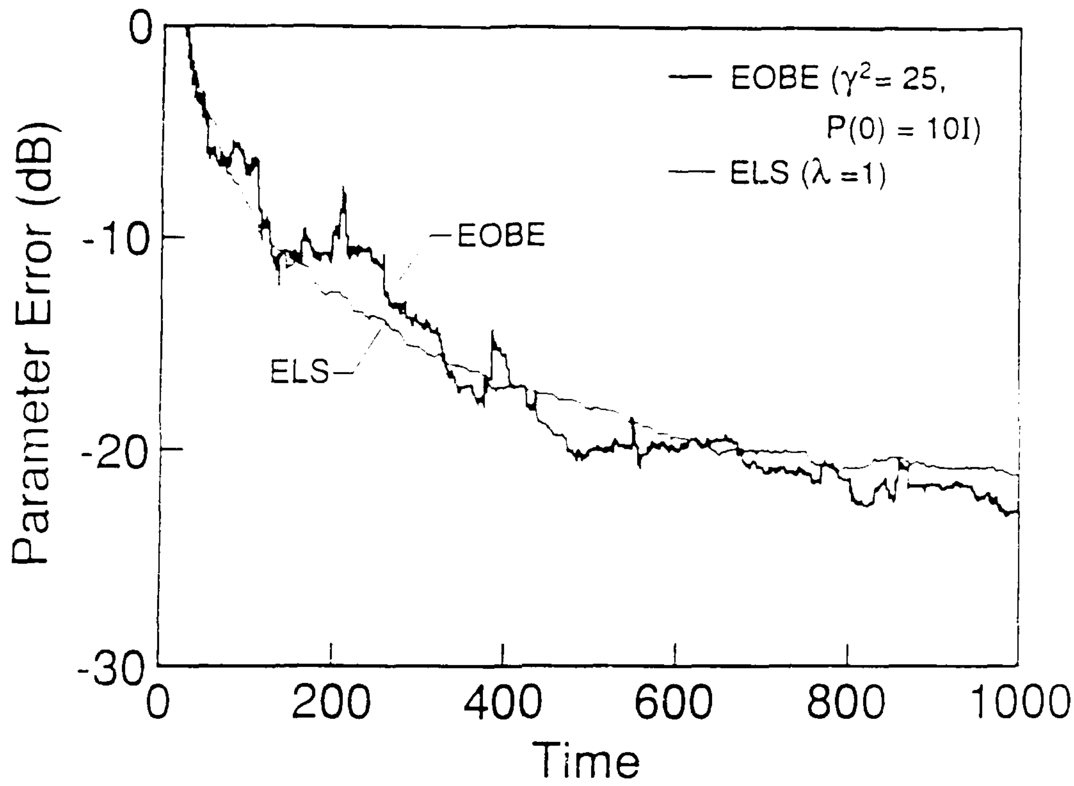

Figure 3.7 Mean-squared MA parameter estimation error - Example 3.2

CHAPTER IV

FINITE PRECISION EFFECTS

4.1 Introduction

The behavior of adaptive filtering algorithms in limited precision environments has attracted an increasing amount of attention lately. The motivation is, perhaps, the discrepancy between the theoretical infinite precision behavior and the actual performance of the algorithms when implemented in hardware or simulated on computers. Finite word-length computations can cause numerical inaccuracy in the results and numerical instability [Cioffi, 1987]. In the case of the RLS algorithm, it has been known for quite some time that the recursion for the covariance matrix inverse is numerically unstable and several factorization methods have been developed to stabilize the recursion [Bierman, 1977]. Numeric instability can cause some of the variables in an adaptive filtering algorithm to become unbounded fairly rapidly, as in the case of fast least-squares algorithms [Cioffi, 1984]. On the other hand, the accumulation of round-off errors can cause the widely used LMS algorithm, and even the stabilized RLS algorithm, to diverge after millions of iterations. In the case of real time signal processing, at a sampling rate of 8 kHz, this amounts to only a few minutes of processing. It is therefore imperative to consider the effects of finite precision computations when analyzing or designing adaptive filter algorithms.

The issue of finite word-length effects in MSPE algorithms has begun to receive attention only recently. In [Walter, 1988], the potential numerical problems which can arise with the exact cone updating (ECU) algorithm are discussed and a robust modification is suggested. In this chapter, finite precision effects in the DHOBE algorithm are studied through simulations and by analyzing the stability properties of the error propagation in the algorithm. Since the update equations of the DHOBE algorithm are very similar to the RLS algorithm, the potential problems which can arise due to the effects of finite word-length in the RLS algorithm are discussed in Section 4.2. In Section 4.3, a sensitivity analysis of the forgetting factor determination formula in the DHOBE algorithm is performed, and a modification is suggested to increase the robustness of the formula. An analysis of the error propagation in the DHOBE algorithm is performed in Section 4.4 by studying the stability properties of two coupled nonlinear difference equations. The equations are shown to be BIBO stable and consequently the error in the estimates is bounded. The effect on the bounding ellipsoid of the errors due to finite word-length computations, in one iteration of the algorithm is studied in Section 4.5. In Section 4.6, the fixed point implementation of the algorithm is described and simulation results are presented which show that the DHOBE algorithm yields consistently good estimates over a large range of word-lengths. In particular, the performance is superior to that of the RLS algorithm for small word-lengths.

4.2 Finite Word-length Effects in the RLS Algorithm

The update equation for the parameter estimates of the RLS algorithm with forgetting factor λ is given by

$$\theta(t) = \theta(t-1) + K(t)\Phi(t)\left[y(t) - \Phi^T(t)\theta(t-1)\right] \tag{4.2.1}$$

where the Kalman gain $K(t)$ is defined by

$$K(t) = \frac{P(t-1)\Phi(t)}{\lambda + \Phi^T(t)P(t-1)\Phi(t)} \tag{4.2.2}$$

and the matrix update equation is given by

$$P(t)=\frac{1}{\lambda}\left[P(t-1)-\frac{P(t-1)\Phi(t)\Phi^T(t)P(t-1)}{\lambda+\Phi^T(t)P(t-1)\Phi(t)}\right] \tag{4.2.3}$$

Due to round-off errors, the subtraction involved in (4.2.3) may cause $P(t)$to become indefinite (neither positive nor negative definite). The resulting least-squares estimates will be incorrect. This sign change, however, does not usually result in overflow [Cioffi, 1987]. The matrix recursion can be stabilized by replacing it with a set of recursions that propagate the upper-diagonal-upper transpose (UDU^T) factorizations of the matrix $P(t)$ [Bierman, 1977]. These recursions ensure that the updating of the diagonal matrix D(t) maintains positive diagonal entries thereby ensuring that $P(t)$ remains positive definite.

The homogeneous difference equation for the error in the estimates of the RLS algorithm is

$$\tilde{\theta}(t)=\theta(t)-\theta^*=[I-K(t)\Phi(t)\Phi^T(t)]\tilde{\theta}(t-1) \tag{4.2.4}$$

It is easy to show that (4.2.4) is exponentially stable if $\lambda < 1$ and P(t) is uniformly bounded [Ljung, 1985] . If $\lambda = 1$, then only asymptotic but not exponential stability can be concluded. Round-off and quantization errors appear as inputs to (4.2.4). If the forgetting factor is equal to unity, then these errors can cause the estimation error to become very large. This observation is confirmed by the detailed statistical analysis in [Ardalan, 1987]. Thus even when the parameters are not time varying, it is advantageous from a numerical point of view to use a forgetting factor less than unity.

Even after stabilization, the RLS algorithm (with $\lambda < 1$) may exhibit long term instability, if the input is spectrally ill conditioned. A heuristic explanation for this phenomenon [Cioffi, 1987], for the case of a FIR adaptive filter adapting to a moving average system model is as follows. Assume that the input x(t) to the adaptive filter is bandlimited, i.e. its Fourier transform satisfies

$$X(e^{j\omega}) = \text{non-zero}, \qquad \text{for } |\omega-\omega_0| < \omega_b$$

and

$$X(e^{j\omega}) = 0 , \qquad \text{for } |\omega - \omega_0| \geq \omega_b$$

If the filter coefficients of the adaptive filter $\theta(t)$ vary very slowly with time, then a Fourier transform $\Theta(e^{j\omega})$ of the filter coefficients could be defined. Then

$$\Theta(e^{j\omega})X(e^{j\omega}) = \text{non-zero}, \qquad \text{for } |\omega - \omega_0| < \omega_b$$

and

$$\Theta(e^{j\omega})X(e^{j\omega}) = 0 , \qquad \text{for } |\omega - \omega_0| \geq \omega_b$$

Over the frequency range $|\omega - \omega_0| \geq \omega_b$, the filter response $\Theta(e^{j\omega})$ can take on very large values without affecting the output error and hence the adaptive filter does not compensate for this growt.. This growth then leads to an overflow of the registers containing the filter coefficients. This unbounded growth can be avoided if a small diagonal constant is added to P(t) at every iteration, or to the diagonal matrix in the UDU^T implementation.

4.3 Sensitivity Analysis

In this section, the effects of small perturbations in the inputs to the updating gain determination formula (2.3.8)-(2.3.14) are studied. If the resulting perturbation in λ_t is small, then the change in the estimates P(t), $\theta(t)$ and $\sigma^2(t)$ is also small. This can be seen by examining the update equations of the algorithm. From (2.3.4), it is easy to see that $P^{-1}(t)$ is a continuous function of λ_t . The partial derivative of $P^{-1}(t)$ with respect to λ_t is

$$\frac{\partial P^{-1}(t)}{\partial \lambda_t} = -P^{-1}(t-1) + \Phi(t)\Phi^T(t)$$

Ass''me that there exist constants M_1 and M_2 such that

$$M_1\mathbf{I} \leq P^{-1}(t) \leq M_2\mathbf{I}$$

and assume further that $\| \Phi(t) \|$ is bounded, i.e. the ARX process is stable and has bounded inputs then

$$\left\| \frac{\partial P^{-1}(t)}{\partial \lambda_t} \right\| < \infty$$

The perturbation $\Delta P^{-1}(t)$ is thus small if the perturbation in λ_t is small. It is shown in the next section that $\Delta P(t) = P(t)\Delta P^{-1}(t)P(t)$. Hence the perturbation in P(t) will be small. Similarly, (2.3.6) shows that $\theta(t)$ is a continuous function of λ_t and that the derivative is bounded provided the eigenvalues of P(t) are upper bounded and $\|\Phi(t)\|$ is bounded. Hence the perturbation in $\theta(t)$ will also be small. The derivative of $\sigma^2(t)$ in (2.3.5) is

$$\frac{d\sigma^2(t)}{d\lambda_t} = \gamma^2 - \sigma^2(t-1) - \frac{(1-\lambda_t)^2 - \lambda_t^2 G(t)}{\left(1-\lambda_t + \lambda_t G(t)\right)^2} \delta^2(t) \qquad (4.3.1)$$

Using the fact that $0 \le \lambda_t \le \alpha < 1$, it is relatively straightforward to show that

$$\frac{-\alpha}{1-\alpha} \le \frac{(1-\lambda_t)^2 - \lambda_t^2 G(t)}{(1-\lambda_t+\lambda_t G(t))^2} \le 1$$

The derivative of $\sigma^2(t)$ is thus bounded, and so the resulting perturbation in $\sigma^2(t)$ is also small.

To b in, a case in which a small change in $\sigma^2(t-1)$ and $\delta^2(t)$ can cause a large change in the updating gain factor λ_t is considered. In the subsequent discussion, the quantities computed with finite precision are denoted by primes. It is also assumed that the perturbations in the inputs to the updating gain factor are small, i.e. there exists a positive $\Delta << 1$ such that

$$|\sigma'^2(t-1) - \sigma^2(t-1)| \le \Delta,\ |\delta'^2(t) - \delta^2 t)| \le \Delta,\ |G'(t) - G(t)| \le \Delta \qquad (4.3.2)$$

Consider the following scenario

$$\sigma^2(t-1) + \delta^2(t) \le \gamma^2 \ \text{ but } \ \sigma'^2(t-1) + \delta'^2(t) > \gamma^2 \qquad (4.3.3)$$

In this case $\lambda_t = 0$, and if $\delta'^2(t) = 0$, then $\lambda'_t = \alpha$. Thus the perturbation in the updating gain factor $\Delta\lambda_t = \lambda'_t - \lambda_t$ can be substantial. It can be shown easily that if (4.3.3) holds then there is a significant difference between λ_t and λ'_t only if $\delta'^2(t)$ is small. However, $\sigma^2(t)$, which is the quantity being minimized, is then marginally affected by the change in λ_t. This is because the derivative in (4.3.1) is close to zero since (4.3.2) and (4.3.3) imply that $\delta^2(t)$ and the difference $(\sigma^2(t-1) - \gamma^2)$ is small. Thus though the perturbation $\Delta\lambda_t$ is large, the difference in the calculated value of $\sigma^2(t)$ will be marginal. The large perturbation $\Delta\lambda_t$ will cause a large perturbation in the matrix P(t), though the perturbation in $\theta(t)$ will be small since $\delta^2(t)$ is small. In general, the formula (2.3.8)-(2.3.14), can cause large perturbations in λ_t, which only marginally affect the resulting value of $\sigma^2(t)$, if the following condition holds for some ε, suitably small

$$|\gamma^2 - \sigma^2(t-1)| \le \varepsilon, \text{ and } \delta^2(t) \le \varepsilon \qquad (4.3.4)$$

In order to make the updating gain formula more robust, it is modified by adding the following additional condition to (2.3.10) – (2.3.14)

$$\text{If } |\gamma^2 - \sigma^2(t-1)| \le \varepsilon \text{ and } \delta^2(t) \le \varepsilon, \text{ for some suitably small } \varepsilon \qquad (4.3.5)$$

$$\text{then} \quad \lambda_t = 0$$

For the modified algorithm if the situation of (4.3.3) occurs then the perturbation in λ_t would be small and the resulting value of $\sigma'^2(t)$ would be almost optimal. The same is true if $\sigma^2(t-1) + \delta^2(t) > \gamma^2$ and $\sigma'^2(t-1) + \delta'^2(t) \le \gamma^2$. However, the modified formula can in some cases still cause a large perturbation $\Delta\lambda_t$. For example if $|\gamma^2 - \sigma^2(t-1)| \le \varepsilon$ and $\delta^2(t) \le \varepsilon$ and $|\gamma^2 - \sigma'^2(t-1)| \le \varepsilon$ and $\delta'^2(t) > \varepsilon$, then $\lambda_t = 0$ and λ'_t could be as large as α. But since $\delta'^2(t)$ is no longer negligible, it is clear from (2.3.5) that the decrease in $\sigma^2(t)$ is not insubstantial. In general, it is easy to show that for cases in which the modified formula yields a large value of $\Delta\lambda_t$, the calculated updating gain factor causes a significant decrease in σ^2. Thus the modification is a compromise between minimizing $\sigma^2(t)$ and reducing the perturbation in the updating gain factor. In contrast using (2.3.8)-(2.3.14), without any modification, could cause large $\Delta\lambda_t$, thus causing large perturbations in $\theta(t)$ and P(t), while decreasing σ^2 only marginally.

The sensitivity of the modified formula to perturbations in its inputs is now evaluated, assuming that the situation of (4.3.4) does not occur for the perturbed and unperturbed algorithms, since it has already been observed that the modified formula can be quite sensitive if (4.3.4) holds .

Theorem 4.1. If the perturbation in $\sigma^2(t-1)$, $G(t)$ and $\delta^2(t)$ is less than Δ, for some small positive number $\Delta < \varepsilon$, where ε is given by (4.3.4), and if (4.3.4) does not hold, then the perturbation in λ_t is of $O(\Delta)$ provided that $\alpha \leq 0.414$.

Proof: The perturbation in the updating gain factor is evaluated assuming that $\sigma^2(t-1) + \delta^2(t) > \gamma^2$ and $\sigma'^2(t-1) + \delta'^2(t) > \gamma^2$. From (2.3.11)-(2.3.14), it can be seen that there are a total of 8 distinct cases in which there can be a perturbation in the updating gain factor:

Case 1: $\delta'^2(t) = 0$ and $G(t) = 1$;

Case 2: $\delta'^2(t)$ and $1+\beta(t)(G(t)-1) > 0$;

Case 3: $\delta'^2(t)=0$ and $1+\beta(t)(G(t)-1) \leq 0$;

Case 4: $G'(t) = 1$ and $G'(t) = 1$;

Case 5: $G'(t) = 1$ and $1+\beta(t)(G(t)-1) > 0$;

Case 6: $G'(t) = 1$ and $1+\beta(t)(G(t)-1) \leq 0$;

Case 7: $1+\beta'(t)(G'(t)-1) > 0$ and $1+\beta(t)(G(t)-1) > 0$;

Case 8: $1+\beta'(t)(G'(t)-1) > 0$ and $1+\beta(t)(G(t)-1) \leq 0$

The perturbation in each case is evaluated now.

Case 1. $\delta'^2(t) = 0$ and $G(t) = 1$:

Since it is assumed that (4.3.4) does not hold, it follows from (2.3.11) that $\lambda'_t = \alpha$. The assumption that the perturbation in the inputs to the formula is less than Δ would imply that $\delta^2(t) \leq \Delta \leq \varepsilon$. Hence $\beta(t) = \dfrac{\gamma^2-\sigma^2(t-1)}{\delta^2(t)} < -1$. Thus

$v_t = \dfrac{1-\beta(t)}{2} > 0.5$. If $\alpha \leq 0.5$, then $\lambda_t = \alpha$, and so the perturbation $\Delta\lambda_t=0$.

Case 2. $\delta'^2(t) = 0$ and $1+\beta(t)(G(t)-1) > 0$:

It has been shown above that $\beta(t) < -1$. This together with $1+\beta(t)(G(t)-1) > 0$, would imply that $G(t) < 2$. Using (2.3.13), it can be shown that when $1+\beta(t)(G(t)-1) > 0$, then $v_t \geq \dfrac{1}{1+\sqrt{G(t)}}$, i.e $v_t \geq 0.414$. Hence if $\alpha \leq 0.414$, then $\lambda_t = \alpha$, and so the perturbation $\Delta\lambda_t = 0$.

Case 3. $\delta'^2(t) = 0$ and $1+\beta(t)(G(t)-1) \leq 0$:

From (2.3.14) $\lambda_t = \alpha$, and hence the perturbation $\Delta\lambda_t=0$.

Case 4: $G'(t) = 1$ and $G'(t) = 1$:

In this case $v'_t = \dfrac{1-\beta'(t)}{2}$ and $v_t = \dfrac{1-\beta(t)}{2}$. Since it is assumed that the situation of (4.3.4) does not occur, the values of $\beta(t)$ and $\beta'(t)$ will differ by an amount greater than $O(\Delta)$ only if $\delta^2(t)$ is small. But then both $\beta(t)$ and $\beta'(t)$ will be large negative numbers and so $\lambda'_t = \lambda_t = \alpha$.

Case 5: $G'(t) = 1$ and $1+\beta(t)(G(t)-1) > 0$:

Let $G(t) = 1+\eta$, where $|\eta| \leq \Delta$. Then from (2.3.13), $v_t = \dfrac{1}{\eta}\left[\sqrt{\dfrac{1+\eta}{1+\eta\beta(t)}} - 1\right]$

Hence $v_t \cong \dfrac{1-\beta(t)}{2} + O(\eta^2)$. Thus as in Case 4, the perturbation will be of $O(\Delta)$.

Case 6: $G'(t) = 1$ and $1+\beta(t)(G(t)-1) \leq 0$:

From (2.3.14), $\lambda_t = \alpha$. Since $|G(t) - G'(t)| \leq \Delta$, therefore either $G(t) = 1+\eta$ or $G(t)= 1-\eta$, where $\eta \leq \Delta$. If $G(t) = 1+\eta$, then $1+\beta(t)(G(t)-1) \leq 0$ would imply that $\beta(t) \leq \frac{-1}{\eta}$, i.e. $\delta^2(t) = O(\eta)$. Hence $\delta'^2(t) \leq \Delta + O(\eta)$ and therefore $\beta'(t) << -1$. Thus $\lambda'_t = \lambda_t = \alpha$. On the other hand if $G(t)= 1-\eta$ then $1+\beta(t)(G(t)-1) \leq 0$ would imply that $\beta(t) \geq \frac{1}{\eta}, > 1$, which contradicts the assumption that $\sigma^2(t-1) + \delta^2(t) > \gamma^2$.

Case 7: $1+\beta'(t)(G'(t)-1) > 0$ and $1+\beta(t)(G(t)-1) > 0$:

An expression for the perturbation Δv_t is obtained by evaluating the partial derivatives of v_t , from (2.3.13), with respect to $\beta(t)$ and $G(t)$. For brevity, the time suffix is dropped in the expressions below

$$\Delta v = \frac{\partial v}{\partial \beta}\Delta\beta + \frac{\partial v}{\partial G}\Delta G$$

where

$$\frac{\partial v}{\partial \beta} = -\frac{\sqrt{G}}{(2[1+\beta(G-1))]^{3/2}}$$

and

$$\frac{\partial v}{\partial G} = \frac{-G + \beta G + \beta - 2\beta G^2 - 1}{2\sqrt{G}(G-1)^2\,[1+\beta(G-1)]^{3/2}}$$

It can now be assumed that there exists a positive η, suitably small such that

$$|G-1| \geq \eta, \text{ and } 1+\beta(G-1) \geq \eta.$$

This is because if, say, $|G-1| < \eta$, then, as in Case 5, $v' \cong v \cong \frac{1-\beta}{2}$ and therefore as discussed in Case 4, $\Delta\lambda$ is small. If $0 < 1+\beta(G-1) < \eta$, then from (2.3.13) $v' >> 1$ and so $\lambda'_t = \lambda_t = \alpha$. The above assumptions along with the assumption that $G(t)$ is bounded ensure that the partial derivatives are bounded and hence there exist K_1 and K_2 such that

$$|\Delta v_t| \leq K_1 |\Delta\beta(t)| + K_2 |\Delta G(t)|$$

Thus if the perturbation $\Delta\beta(t)$ and $\Delta G(t)$ is of $O(\Delta)$ then the perturbation $\Delta\lambda_t$ is also of $O(\Delta)$.

Case 8: $1+\beta'(t)(G'(t)-1) > 0$ and $1+\beta(t)(G(t)-1) \leq 0$:

In this case, $\lambda_t = \alpha$. If $|G'(t)-1| \leq \eta$ for some small number η, then as in Case 6,

$$\beta'(t) \leq \frac{-1}{\eta} \text{ and as in Case 5, } \nu'_t \cong \frac{1-\beta'(t)}{2} >> 1, \text{ and hence } \lambda'_t = \alpha.$$

If G(t) differs sufficiently from unity then either $1+\beta'(t)(G'(t)-1)$ is very small or else $\beta'(t)$ and $\beta(t)$ differ greatly. If $1+\beta'(t)(G'(t)-1)$ is small then from (2.3.13) $\nu'_t > \alpha$, and so $\lambda'_t = \alpha$. On the other hand $\beta'(t)$ and $\beta(t)$ differ substantially only when $\delta^2(t)$ is small (assuming (4.3.4) does not hold). Then by the same argument as in Case 2, $\lambda'_t = \alpha$, if $\alpha < 0.414$.

$\nabla\nabla\nabla$

4.4 Error Propagation in the DHOBE Algorithm

The error propagation properties of the DHOBE algorithm are analyzed by focusing on the propagation of a single error in $\theta(t)$ and $P(t)$ to future instants. Assume that at time instant t_0 there is a perturbation due to round-off error in the estimates $\theta(t_0)$ and $P(t_0)$, so that $\theta'(t_0) = \theta(t_0) + \Delta\theta(t_0)$ and $P'(t_0) = P(t_0)+\Delta P(t_0)$, where, the primed quantities are the perturbed ones. The problem considered here is the effect of these errors on the estimates $\theta'(t)$ and $P'(t)$ at $t > t_0$, assuming that the computations are performed with infinite precision. Similar studies have been performed by Ljung and Ljung [Ljung, 1985] in their investigation of the error propagation properties of RLS algorithms. Though the update equations of the DHOBE algorithm are similar to those of the RLS algorithm, the presence of the updating gain factor, which is a discontinuous function of the estimates, complicates the analysis. In particular, the results on perturbed linear differential equations (used in [Ljung, 1985]) cannot be applied. Error propagation in the

DHOBE algorithm is investigated by performing a first order perturbation analysis of two coupled nonlinear difference equations. The analysis yields coupled difference equations whose homogeneous parts are exponentially stable. An upper bound on the error in the estimates due to finite precision computations is given by the following result:

Theorem 4.2. If the following assumptions hold:

(i) The matrix P(t) is well conditioned, i.e. there exist positive η_1 and η_2 such that

$$0 < \eta_1 \mathbf{I} < P(t) < \eta_2 \mathbf{I} \quad \text{for all } t \tag{4.4.1}$$

(ii) The ARX process is stable and has bounded inputs,

(iii) The unperturbed algorithm yields bounded prediction errors,

(iv) For some integer M, if the unperturbed algorithm has M updates in an interval of time, then the perturbed version updates at least once in that interval,

(v) At the updating instants of the perturbed algorithm, a lower bound ρ is set for the updating gain factor λ_t', where ρ is a suitably small positive number.

Then there exist constants m_1 and h which depend on the bounds on the prediction error of the unperturbed algorithm and the inputs and outputs of the process such that the error between the perturbed and unperturbed quantities at the updating instants $\{t_k\}$ of the perturbed algorithm is bounded as

$$\|\Delta P(t_k)\| \leq \left(\frac{\eta_2}{\eta_1}\right)^2 (1-\rho)^{k/M-1} \|\Delta P(0)\| + \eta_2^2 m_1 \max_{1\leq u\leq k} \Delta\lambda_{t_u} M \frac{1-(1-\rho)^{\lfloor k/M \rfloor}}{\rho} \tag{4.4.2}$$

$$\|\Delta\theta(t_k)\| \leq (1-\rho)^{k/M-1} \|\Delta\theta(t_0)\| + \eta_2 h \max_{1\leq j\leq k} |\Delta\lambda_{t_j}| \frac{M}{\rho} [1-(1-\rho)^{\lfloor k/M \rfloor}] +$$

$$+ h \frac{\eta_2}{\eta_1} \max_{1\leq j\leq k} \|\Delta P(t_j)\| \tag{4.4.3}$$

where η_1 and η_2 are as in (4.4.1).

Proof :

Define $R(t) = P^{-1}(t)$. Then

$$R(t) = (1-\lambda_t)\, R(t-1) + \lambda_t \Phi(t)\, \Phi^T(t) \tag{4.4.4}$$

Assume that R(t) and θ(t) are perturbed by an infinitesimal amount at time instant t_0. The perturbed quantities at any time instant t, will satisfy

$$R'(t) = (1-\lambda_t')\, R'(t-1) + \lambda_t' \Phi(t)\, \Phi^T(t) \tag{4.4.5}$$

where, as before, the perturbed quantities are denoted by primes. Subtracting (4.4.4) from (4.4.5) it follows that

$$\Delta R(t) = (1-\lambda_t')\, \Delta R(t-1) - \Delta\lambda_t\, [\Phi(t)\, \Phi^T(t) - R(t-1)] \tag{4.4.6}$$

where

$$\Delta R(t) = R'(t) - R(t) \text{ and } \Delta\lambda_t = \lambda_t' - \lambda_t, \tag{4.4.7}$$

The difference equation (4.4.6) can be expressed as

$$\Delta R(t) = \prod_{i=t_0+1}^{t} (1-\lambda_i')\Delta R(t_0) + \Delta\lambda_t(\Phi(t)\Phi^T(t) - R(t-1)) + I(t) \tag{4.4.8}$$

where

$$I(t) = \sum_{j=t_0+1}^{t-1} \prod_{i=j+1}^{t} (1-\lambda_i')\Delta\lambda_j[\Phi(t-j)\Phi^T(t-j) - R(t-j-1)] \tag{4.4.9}$$

The summation in (4.4.9) can be taken over the subsequence $\{t_u, u = 1, 2, ..\}$ of instants at which updates are performed for either the perturbed or the unperturbed algorithm. This is because at all other instants i , $\Delta\lambda_i = 0\text{-}0 = 0$. Also, by the assumptions (i) and (ii) of Theorem 3.2, there exists a constant $m_1 > 0$, such that for all t,

$$\|\Phi(t)\, \Phi^T(t) - R(t-1)\| \leq m_1 \tag{4.4.10}$$

Thus at any instant $t_k \in \{t_u\}$

$$\|I(t_k)\| \leq m_1 \sum_{u=1}^{k-1} \prod_{r=u+1}^{k} (1-\lambda'_{t_r})\, |\Delta\lambda_{t_u}| \tag{4.4.11}$$

Now assumption (iv) of the theorem implies that every M consecutive elements of the subsequence $\{t_u\}$ contains at least one instant at which the perturbed algorithm performs an update. Thus at least $\lfloor k/M \rfloor$,(where $\lfloor . \rfloor$ denotes integer part), updates of the perturbed algorithm have occurred at time instant t_k. Then using assumption (v), (4.4.11) can be expressed as

$$\| I(t_k) \| \leq m_1 \max_{1 \leq u \leq k-1} |\Delta \lambda_{t_u}| [M-1 + M (1-\rho + (1-\rho)^2 \ldots + (1-\rho)^{\lfloor k/M \rfloor})] \qquad (4.4.12)$$

For example if k=10 and M=3, then in the worst possible case, i.e. one in which the right hand side of (4.4.11) would be the largest, there would be only one update in the perturbed algorithm for every three instants of $\{t_u\}$. Hence from (4.4.11) an upper bound on $\| I(t_k) \|$ would be

$$\| I(t_k) \| \quad \leq \quad m_1 \max_{1 \leq i \leq 9} |\Delta \lambda_{t_i}| \; [1+1+3(1-\rho) + 3(1-\rho)^2 + 2(1-\rho)^3]$$

Upper bounding the first and second terms in the right hand side of (4.4.8) and using (4.4.12) then yields the following expression for the norm of the error in the matrix R(t) at instants t_k which are updating instants for either the perturbed or the unperturbed algorithm

$$\| \Delta R(t_k) \| \leq \quad (1-\rho)^{k/M-1} \; \| \Delta R(t_0) \| + m_1 \max_{1 \leq u \leq k} | \Delta \lambda_{t_u} | \, M \, \frac{1-(1-\rho)^{\lfloor k/M \rfloor}}{\rho} \qquad (4.4.13)$$

Thus the perturbation in $R(t_k)$ is bounded. Note that no approximations were required to obtain (4.4.13). In order to obtain an upper bound on the error in P(t), a first order analysis is performed. It is assumed that

$\| \Delta R(t) \| = O(\varepsilon)$, with $\varepsilon << \lambda_{min}(R(t))$ where λ_{min} refers to the minimum eigenvalue.

Then

$$P'(t) = R'^{-1}(t) = (R(t)+\Delta R(t))^{-1} = R^{-1}(t) - R^{-1}(t) \, \Delta R(t) \, R^{-1}(t) + \Delta_1 \qquad (4.4.14)$$

where $\| \Delta_1 \| = O(\varepsilon^2)$. Neglecting $O(\varepsilon^2)$ terms yields

$$\Delta P(t) = P'(t) - P(t) = P(t)\Delta R(t)P(t) \tag{4.4.15}$$

Thus

$$\| \Delta P(t) \| \le \eta_2^2 \| \Delta R(t) \| \tag{4.4.16}$$

Similarly, neglecting $O(\varepsilon^2)$ terms yields

$$\| \Delta R(t) \| \le 1/\eta_1^2 \| \Delta P(t)\| \tag{4.4.17}$$

where η_1 and η_2 are defined in (4.4.1). Finally, using (4.4.16) and (4.4.17) in (4.4.13) yields (4.4.2).

From (2.3.6) and (2.3.7), the time recursions for $\theta(t)$ and $\theta'(t)$ can be expressed as

$$\theta(t) = [\, \mathbf{I} - \lambda_t\ L(t)\, \Phi^T(t)\,] + \lambda_t\, L(t)\, y(t) \tag{4.4.18}$$

and

$$\theta'(t) = [\, \mathbf{I} - \lambda_t'\ L'(t)\, \Phi^T(t)\,] + \lambda_t'\, L'(t)\, y(t) \tag{4.4.19}$$

where

$$L(t) = P(t)\, \Phi(t) \tag{4.4.20}$$

Subtracting (4.4.18) from (4.4.19) and performing some manipulations yields

$$\Delta\theta(t) = [\mathbf{I} - \lambda_t' L'(t)\Phi^T(t)]\, \Delta\theta(t-1) + [\Delta\lambda_t\, L(t) + \lambda_t'\Delta L(t)]\, [y(t) - \Phi^T(t)\theta(t-1)] \tag{4.4.21}$$

or

$$\Delta\theta(t) = [\mathbf{I} - \lambda_t' L'(t)\Phi^T(t)]\, \Delta\theta(t-1) + J_1(t) + J_2(t) \tag{4.4.22}$$

where

$$J_1(t) = \Delta\lambda_t\, L(t)\, (y(t) - \Phi^T(t)\theta(t-1)) = \Delta\lambda_t\, L(t)\, \delta(t) \tag{4.4.23a}$$

and

$$J_2(t) = \lambda_t'\, \Delta L(t)\, (y(t) - \Phi^T(t)\theta(t-1)) = \lambda_t'\Delta L(t)\, \delta(t) \tag{4.4.23b}$$

Since

$$[\mathbf{I} - \lambda_t' L'(t)\Phi^T(t)] = (1 - \lambda_t')\, P'(t)\, P'^{-1}(t-1) \tag{4.4.24}$$

Therefore (4.4.22) can be expressed as

$$\Delta\theta(t) = (1 - \lambda_t')\, P'(t)\, P'^{-1}(t-1)\, \Delta\theta(t-1) + J_1(t) + J_2(t) \tag{4.4.25}$$

Solving the difference equation (4.4.25) yields

$$\Delta\theta(t) = \prod_{i=t_0+1}^{t} (1-\lambda_i') P'(t) P'^{-1}(t_0)\Delta\theta(t_0) + J_1(t) + J_2(t) + J_3(t) + J_4(t) \qquad (4.4.26)$$

where

$$J_3(t) = \sum_{j=t_0+1}^{t-1} \prod_{i=j+1}^{t} (1-\lambda_i') P'(t) P'^{-1}(j) \Delta\lambda_j L(j) \delta(j) \qquad (4.4.27a)$$

and

$$J_4(t) = \sum_{j=t_0+1}^{t-1} \prod_{i=j+1}^{t} (1-\lambda_i') P'(t) P'^{-1}(j) \lambda_j' \Delta L(j) \delta(j) \qquad (4.4.27b)$$

Since the perturbation $\Delta R(t)$ is of order ε, from (4.4.15) the perturbation $\Delta P(t)$ is of the same order. In order to facilitate the analysis, it is assumed that $\Delta\lambda_t = O(\varepsilon)$. Then substituting for $L(j)$ from (4.4.20) and neglecting $O(\varepsilon^2)$ terms yields

$$J_3(t) = \sum_{j=t_0+1}^{t-1} \prod_{i=j+1}^{t} (1-\lambda_i') P(t) \Delta\lambda_j \Phi(j) \delta(j) \qquad (4.4.28)$$

Hence

$$\|J_3(t)\| \le \eta_2 \max_{t_0 \le j \le t-1} [\, \|\Phi(j)\| \, |\delta(j)| \,] \sum_{j=t_0+1}^{t-1} \prod_{i=j+1}^{t} (1-\lambda_i') \; \Delta\lambda_j \qquad (4.4.29)$$

Assumptions (ii) and (iii) ensure that there exists a constant $h > 0$, such that

$$\|\Phi(j)\| . |\delta(j)| \le h < \infty$$

Now , as before, the upper bound (4.4.29) is evaluated at the updating instants of the perturbed and the unperturbed algorithm. The summation of products in (4.4.29) can be upper bounded as in (4.4.13) by

$$\sum_{j=t_0+1}^{t-1} \prod_{i=j+1}^{t} (1-\lambda_i') \Delta\lambda_j \le \max_{1 \le u \le k-1} |\Delta\lambda_{t_u}| \, [\, M - 1 + M(1-\rho + (1-\rho)^2 + .. + (1-\rho)^{k/M_{\cdot}})$$

Hence

$$\|J_1(t_k)\| + \|J_3(t_k)\| \leq \quad \eta_2 h \max_{1\leq j\leq k} |\Delta\lambda_j| \, M \frac{1-(1-\rho)^{\lfloor k/M \rfloor}}{\rho} \tag{4.4.30}$$

Since $\Delta L(j) = \Delta P(j)\Phi(j)$, after neglecting second order terms (4.4.27b) becomes

$$J_4(t) = \sum_{j=t_0+1}^{t-1} \prod_{i=j+1}^{t} (1-\lambda_i') \, \lambda_j' \; P(t)\, P^{-1}(j)\, \Delta P(j)\, \Phi(j)\delta(j) \tag{4.4.31}$$

Thus

$$\|J_2(t)\| + \| J_4(t)\| \leq \; h\frac{\eta_2}{\eta_1} \max_{1\leq j\leq t} \| \Delta P(j) \| \left[\sum_{j=t_0+1}^{t-1} \lambda_j' \prod_{i=j+1}^{t} (1-\lambda_i') + \; \lambda_t' \right] \tag{4.4.32}$$

In Appendix 4A, it is shown that the term in the square brackets in (4.4.32) is less than unity and so combining (4.4.26), (4.4.30) and (4.4.32) yields

$$\|\Delta\theta(t_k)\| \; \leq \; (1-\rho)^{k/M-1} \| \Delta\theta(t_0) \| + \eta_2 h \max_{1\leq j\leq k} |\Delta\lambda_{t_j}| \frac{M}{\rho} [1-(1-\rho)^{\lfloor k/M \rfloor}] +$$

$$h\frac{\eta_2}{\eta_1} \max_{1\leq j\leq k} \|\Delta P(t_j)\|$$

Thus the perturbation in the parameter estimates is also bounded.

$\nabla\nabla\nabla$

Remarks

(1) Assumption (iv) is a technical artifice which facilitates the analysis. It is highly unlikely that it would ever be violated. Violation of this assumption would imply that the squared prediction error for the perturbed algorithm is upper bounded by γ^2 for large durations of time. Then, provided the input and noise sequences are sufficiently rich and uncorrelated, it is easy to show that $\| \theta'(t) - \theta^* \|^2 \leq O(\gamma^2)$ for those intervals of time.

(2) Assumption (v) is a technical device required to ensure that the homogeneous parts of (4.4.2) and (4.4.3) are exponentially stable. If $\rho \leq 0.001$, then in practice the values of λ_t at the updating instants will usually be larger than ρ.

(3) Though the analysis of error propagation has ignored the effect of round-off errors in computations, since the homogeneous parts of (4.4.2) and (4.4.3) are exponentially stable, the errors at any time instant due to round-off errors created at previous time instants would be bounded [Ljung, 1985].

4.5 Finite Precision Effects on the Bounding Ellipsoid

In this section, the effect of round-off errors (in one iteration) on the resulting bounding ellipsoid is studied. More specifically, we ask the question - If $\theta^* \in E_{t-1}$, can errors in the computation of E_t (i.e. computation of $\theta(t)$, $P(t)$ and $\sigma^2(t)$) cause $\theta^* \notin E_t$.

Define $\tilde{\theta}(t) = \theta(t) - \theta^*$. Then from (2.3.6)

$$\tilde{\theta}(t) = \tilde{\theta}(t-1) + \lambda_t P(t)\Phi(t)\delta(t) + \Delta_1 \tag{4.5.1}$$

where Δ_1 is the round-off error. Similarly from (2.3.4)

$$P^{-1}(t) = (1-\lambda_t)\, P^{-1}(t-1) + \lambda_t\, \Phi(t)\Phi^T(t) + \Delta_2 \tag{4.5.2}$$

and

$$\sigma^2(t) = (1-\lambda_t)\, \sigma^2(t-1) + \lambda_t\, \gamma^2 - \frac{\lambda_t (1-\lambda_t)\, \delta^2(t)}{1-\lambda_t + \lambda_t\, \Phi^T(t)P(t-1)\Phi^T(t)} + \Delta_3 \tag{4.5.3}$$

Define

$$A_t = \tilde{\theta}(t-1) + \lambda_t P(t)\Phi(t)\delta(t) \tag{4.5.4}$$

and

$$B_t = (1-\lambda_t)\, P^{-1}(t-1) + \lambda_t\, \Phi(t)\Phi^T(t) \tag{4.5.5}$$

Then, after neglecting second and higher order terms in Δ_1 and Δ_2, it can be shown that

$$V(t) = A_t^T B_t A_t + A_t^T \Delta_2 A_t + \Delta_1^T B_t A_t + A_t^T B_t \Delta_1 \tag{4.5.6}$$

where

$$V(t) = \tilde{\theta}(t)\, P^{-1}(t)\, \tilde{\theta}(t) \tag{4.5.7}$$

Expanding $A_t^T B_t A_t$ as in Section 3.3 and using (4.5.3) yields

$$V(t)-\sigma^2(t) = (1-\lambda_t)\,[V(t-1)-\sigma^2(t-1)] + \lambda_t\,[v^2(t)-\gamma^2]$$
$$+\, 2\Delta_1{}^T B_t A_t + A_t{}^T\Delta_2 A_t + \Delta_3 \qquad (4.5.8)$$

From the definition of E_t it is clear that $\theta^* \in E_t$ iff $V(t) \leq \sigma^2(t)$. Thus if the errors Δ_1, Δ_2, and Δ_3 are large enough, it is possible that $\theta^* \notin E_t$. A sufficient condition for $\theta^* \in E_t$ is

$$|\, 2\Delta_1{}^T B_t A_t + A_t{}^T\Delta_2 A_t \,| \leq \lambda_t\,[\gamma^2 - v^2(t)] \qquad (4.5.9)$$

In case $\lambda_t = 0$, then since no update occurs $\theta^* \in E_t$ automatically. The condition (4.5.9) shows that if the errors due to finite word-length computations are small enough then $\theta^* \in E_t$ and furthermore, by setting γ^2 higher than the actual bound on the noise, the robustness of the algorithm to finite precision effects can be increased.

4.6 Simulation Studies

Simulation Setup. A fixed point implementation of the OBE algorithm was simulated by performing the operations in integer arithmetic. The input and output observations, which are generated as floating point numbers, are converted to integers by the formula

$$x_{quant} = \begin{cases} \mathrm{INT}(\, x.\, 2^{ibit} + 0.5)\,, & x > 0 \\ \mathrm{INT}(\, x.\, 2^{ibit} - 0.5)\,, & x \leq 0\,. \end{cases}$$

where *ibit* is the number of bits assigned for the integer representation of the fractional part of the real number x. In the simulations, since an integer is stored in 32 bits, all registers and word sizes are 32 bits. Multiplication is performed by forming the product in a 48-bit word, scaling down by 2^{-ibit}, and then rounding off to the nearest integer. Inner products are formed similarly by accumulating the products in a 48-bit word, scaling down and then rounding off.

In order to minimize the effect of round-off errors and finite word-length storage, the recursions of the DHOBE algorithm are implemented as shown below

$$\theta(t) = \theta(t-1) + K(t)\delta(t)$$

$$\delta(t) = y(t) - \theta^T(t-1)\,\Phi(t)$$

$$K(t) = \frac{\lambda_t P(t-1)\Phi(t)}{1-\lambda_t+\lambda_t G(t)}$$

$$G(t) = \Phi^T(t)\,P(t-1)\,\Phi(t)$$

$$P(t) = \frac{1}{1-\lambda_t}\left[\, I - K(t)\Phi^T(t)\,\right] P(t-1)$$

$$\sigma^2(t) = (1-\lambda_t)\,\sigma^2(t-1) + \lambda_t\,\gamma^2 - \frac{\lambda_t\,(1-\lambda_t)\,\delta^2(t)}{1-\lambda_t + \lambda_t\,\Phi^T(t)P(t-1)\Phi^T(t)}$$

Notice that the only difference between these equations and (2.3.5)-(2.3.7) is that the parameter estimate update is performed using P(t-1) and hence errors introduced in the formation of P(t) do not affect $\theta(t)$.

The upper bound α on the forgetting factor, has to be chosen with care in the fixed point implementation of the OBE and EOBE algorithms. If α is chosen greater than 0.1, then the elements of the matrix P often increase rapidly in magnitude and overflows can occur. The reason for this is that in the initial stages, the optimum value of the forgetting factor λ equals α fairly often. Consequently, since $1-\lambda$ appears in the denominator of (2.3.7), the magnitude of the elements of P can increase and cause overflows. On the other hand, if α is chosen too small then the algorithm takes more iterations to converge and the number of updates increases. A value of $\alpha = 0.1$ was found to yield a satisfactory convergence rate and inhibit overflows in the update equation for P(t).

In addition to α, the initial value $\sigma^2(0)$ has to be chosen small enough to prevent overflows in the subsequent calculations of λ. This is because if, at any time t, $\sigma^2(t-1)$ is large and $\delta^2(t)$ is small then $\beta = (\gamma^2-\sigma^2(t-1))/\,\delta^2(t)$ can become a very large negative number and the product $\beta(G-1)$ can overflow. However, if overflows can be detected and a saturation value is used for β, then the calculation of λ will not be affected. Since β is negative and large in magnitude, $1+\beta(G-1)$ is a large positive or negative number depending on whether G is greater than or less than unity. In case $1+\beta(G-1)$ is positive, then it can be seen from (4) that v_t is greater than unity, and consequently $\lambda = \alpha$. On the other hand if $1+\beta(G-1)$ is negative then $\lambda = \alpha$ from (4). Thus large values of $\sigma^2(0)$ can be used if care is taken to account for overflows in the algorithm for calculating λ. Alternatively, the formula can just set $\lambda = \alpha$ if $\delta^2(t)$ is smaller than a suitably small number. the In the simulations, the initial (unquantized) value $\sigma^2(0) = 100$.

For the RLS algorithm, the initial value P(0) is also important. Since the bias in the estimates is inversely proportional to P(0), P(0) should be large. However if P(0) is too large, then finite word-length effects can cause the Kalman gain vector K to overflow, and the parameter estimates to grow exponentially in the initial stage [Ardalan, 1987]. Therefore a compromise value - $P(0) = 10\,\mathbf{I}$ was chosen.

Simulation Results

The performance of the DHOBE algorithm is compared to that of the RLS and the exponentially weighted recursive least-squares (EWLS) algorithms, for three different processes. In all the cases, the noise sequence $\{v(t)\}$ and the input sequence $\{u(t)\}$ are generated by a pseudo-random number generator with a uniform probability distribution in [-1.0,1.0]. The upper bound γ^2 is set equal to 1.0. The parameter estimates are obtained by applying the DHOBE, RLS and EWLS (with weighting factor $\lambda = 0.99$) to 1000 point data sequences. Ten runs of the algorithm are performed on the same model but with different noise sequences. The number of bits used for the fractional part, *ibit*, is varied from 16 down to 6 bits and the average of the parameter error $\|\theta(1000)-\theta^*\|^2$ is computed for each value of *ibit*.

Example 4.1(Fig.4.1) AR(5) process

$$y(t) = -0.326\,y(t-1) - 0.427\,y(t-2) - 0.717y(t-3) - 0.288\,y(t-4) - 0.399\,y(t-5) + v(t)$$

It can be seen from Fig. 4.1 that the performance of the DHOBE algorithm appears to be constant as the number of bits varies from 16 to 8. In contrast, the performance of the RLS and EWLS algorithms degrades for *ibit* ≤ 8. For the RLS algorithm the P matrix did not remain positive definite in many runs for *ibit* ≤ 8. For the EWLS algorithm, this happened for *ibit* ≤ 12.

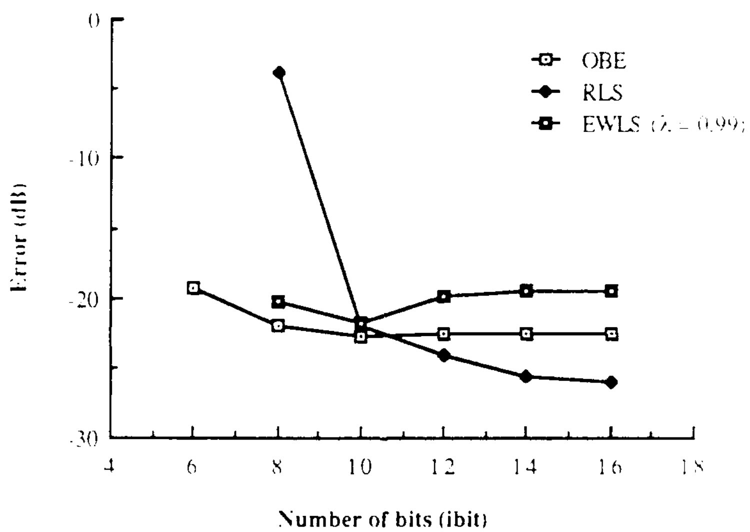

Figure 4.1 Average parameter estimation error for the DHOBE and RLS algorithms for an AR(5) process

Example 4.2 (Fig. 4.2) ARX(2,3) process

$$y(t) = 1.6y(t-1) - 0.83y(t-2) + 0.14u(t) + u(t-1) + 0.16u(t-2) + v(t)$$

As before, the average tap error of the DHOBE algorithm appears constant as *ibit* varies from 16 to 8 bits. The P matrix became negative definite for *ibit* = 6.The RLS and EWLS algorithms do not work well for *ibit* ≤ 10. In fact P became indefinite for *ibit* ≤ 14, in the EWLS case.

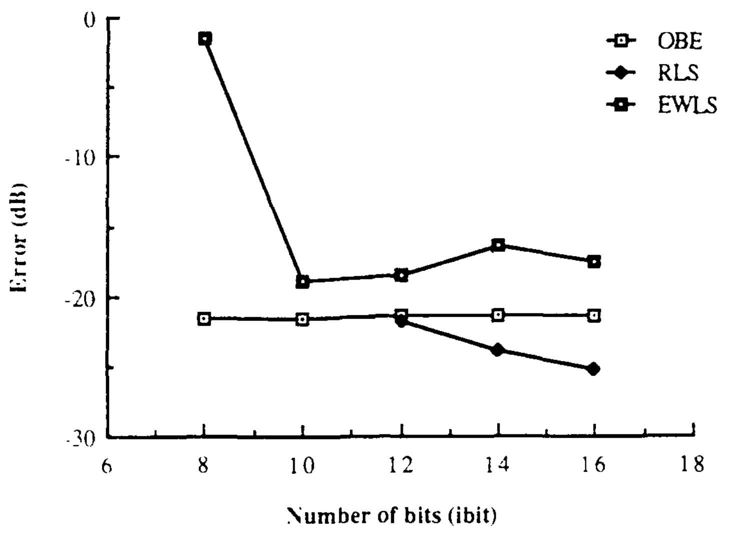

Figure 4.2 Average parameter estimation error for the DHOBE and RLS algorithms for an ARX(2,3) process

Example 4.3 (Fig. 4.3) ARX(10,10) process

The DHOBE algorithm worked well for *ibit* $\geq$ 12. However for smaller values, P became indefinite and overflows occurred. For the RLS case, P became indefinite for *ibit* $\leq$ 14. In order to study the performance of the DHOBE algorithm at smaller word-lengths, a UDU' factorization of the P matrix was performed. The DHOBE update equations are identical to the update equations of the weighted RLS algorithm with weight $\alpha_t = \lambda_t$, and forgetting factor $\lambda(t) = (1-\lambda_t)$ and hence the UDU' form of the DHOBE can be easily developed [Ljung, 1983, pg. 334] . The UDU' form of the DHOBE algorithm is then compared to the UDU' form of the RLS algorithm. The simulation results show that for larger word sizes, the performance of the RLS algorithm is superior. For smaller values of ibit, the average parameter estimation error is about the same for both the DHOBE and the RLS algorithms.

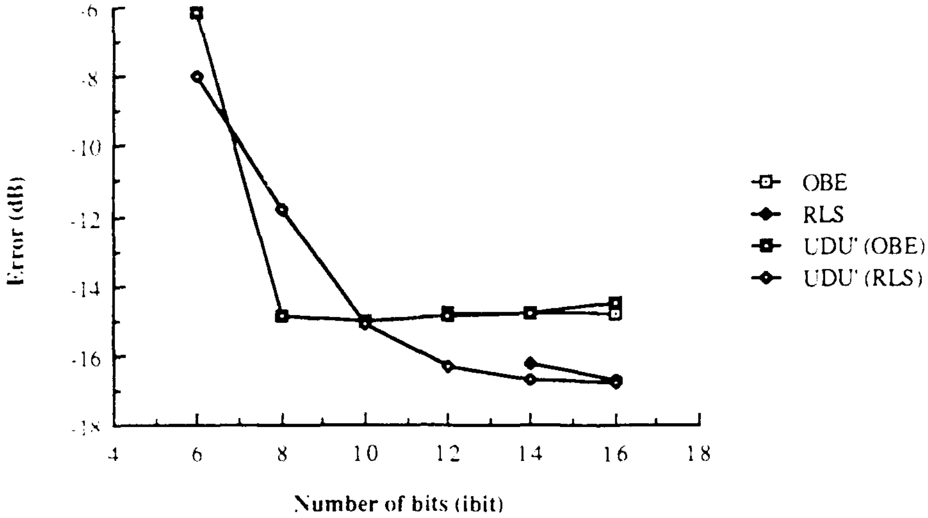

Figure 4.3 Average parameter estimation error for the DHOBE and RLS algorithms for an ARX(10,10) process

Discussions

Example 4.3 shows that the performance of the UDU' versions of DHOBE and RLS algorithms is comparable at smaller word-lengths. The superior performance of the straightforward implementation of the DHOBE algorithm, as compared to the RLS or EWLS algorithms at smaller word-lengths is therefore primarily due to the superior numerical properties of the recursion for the matrix P(t).

The update equation for the RLS algorithm with a forgetting factor λ is

$$P(t) = \left[I - \frac{P(t-1)\Phi(t)\Phi^T(t)}{\lambda + \Phi^T(t)P(t-1)\Phi(t)} \right] \frac{P(t-1)}{\lambda} \tag{4.6.1}$$

The corresponding equation for the DHOBE algorithm can be rewritten as

$$P(t) = \left[I - \frac{P(t-1)\Phi(t)\Phi^T(t)}{\frac{1-\lambda_t}{\lambda_t} + \Phi^T(t)P(t-1)\Phi(t)} \right] \frac{P(t-1)}{1-\lambda_t} \tag{4.6.2}$$

Since $1-\lambda_t$ plays the same role in the DHOBE algorithm as does λ in the RLS algorithm, the only difference between (4.6.1) and (4.6.2) is that the factor $(1-\lambda_t)/\lambda_t$ appears in the denominator of the term within braces in (4.6.2) as opposed to the corresponding term λ in (4.6.1). The degradation of performance occurs primarily because the term within braces becomes indefinite on account of round-off errors. Since λ_t is usually much smaller than unity, the term which is being subtracted from the identity matrix in (4.6.2) is much smaller than the one in (4.6.1). Thus P(t) in the RLS algorithm has a greater tendency to become indefinite than the P(t) in the DHOBE algorithm. This observation has been confirmed by examining the eigenvalues of P(t), for runs in which the RLS algorithm performed poorly.

CHAPTER V

TRACKING ANALYSIS

5.1 Introduction

Performance analysis of adaptive filtering algorithms is usually performed by assuming that the unknown system which is being modeled is time-invariant. However, in actual practice, adaptive filters are often used in time varying environments, and hence it is important to characterize the performance of these algorithms when the system model parameters can vary with time. A considerable amount of attention has been paid to this problem in the adaptive filtering literature, with analysis being performed mainly for the LMS and RLS algorithms, with varying amounts of r[illegible]. See for example [Widrow, 1976], [Benveniste, 1982], [Eleftheriou, 1986], [Rao, 1988], [Gunnarsson, 1989]. It was mentioned in Chapter II that the incorporation of a forgetting factor in the DHOBE algorithm is expected to enhance its tracking performance. In this chapter, the tracking characteristics of the algorithm are studied in some detail. The analysis is, in some ways, simplified by the assumption of bounded noise. However, as in the previous chapter, the presence of the data dependent updating factor complicates any derivation of expressions for the tracking error (the error between the parameter estimates and the time varying parameters). Time varying parameter estimation using the LMS algorithm has been analyzed using a deterministic approach in [Anderson, 1983]. The exponential stability of the homogeneous difference equation for the error between the parameter estimates and the true parameter is used to show bounded parameter estimation error when the parameters are slowly time varying. This approach could be used for the DHOBE algorithm also but the analysis is complicated by the possibility of a zero updating gain factor, though, an exponential stability type property could still be formulated. However, for the DHOBE algorithm (and other membership-set estimation algorithms) it is also equally important to ensure that the time varying true parameters $\{\theta^*(t)\}$ are contained in the bounding ellipsoids $\{E_t\}$. So instead of characterizing tracking in terms of the parameter estimation error, we seek conditions which will ensure that the time varying true parameter vector is contained in the bounding ellipsoids. This will also guarantee bounded parameter estimation error. Section 5.2 discusses some necessary and sufficient conditions for the ellipsoids to contain $\theta^*(t)$. It is also shown that if in case a jump in the true parameter vector θ^* causes it to fall outside the bounding ellipsoid, then provided the jump is not too large, the bounding ellipsoids will move towards θ^* and eventually enclose θ^*. In Section 5.3, a rescue device is proposed, which will guarantee the existence of ellipsoids in the face of large parameter variations. Simulation results are presented in Section 5.4 which show that the DHOBE algorithm is able to track slow and abrupt variations in the parameters. The tracking performance, in terms of parameter estimation error, is in many cases comparable to the RLS algorithm with forgetting factor.

5.2 Necessary and Sufficient Conditions for Tracking

As mentioned above, tracking in the context of bounding ellipsoidal parameter estimation, will mean ensuring that the time varying true parameter vector is contained in the bounding ellipsoid. The theorems below present conditions for parameter tracking.

Theorem 5.1 A sufficient condition for $\theta^*(t) \in E_t$ is

$$(\theta^*(t) - \theta(t-1))^T P^{-1}(t-1)(\theta^*(t) - \theta(t-1)) \leq \sigma^2(t-1) \tag{5.2.1}$$

Proof: If $\theta^*(t) \in E_{t-1}$ then since $\theta^*(t) \in S_t$ and $E_t \supseteq E_{t-1} \cap S_t$, it follows that $\theta^*(t) \in E_t$. And from (2.3.1), $\theta^*(t) \in E_{t-1}$ is equivalent to (5.2.1).

Theorem 5.2 $\theta^*(t) \in E_t$ if and only if

$$(\theta^*(t)-\theta(t-1))^T P^{-1}(t-1)(\theta^*(t)-\theta(t-1)) \le \sigma^2(t-1)+\frac{\lambda_t}{1-\lambda_t}\left(\gamma^2 - v^2(t)\right) \tag{5.2.2}$$

where $v(t)$ is the noise term from (2.2.1).

Proof: Subtracting $\theta^*(t)$ from both sides of (2.3.6) yields

$$\theta(t)-\theta^*(t) = \theta(t-1)-\theta^*(t)+\lambda_t P(t)\Phi(t)\delta(t) \tag{5.2.3}$$

Then using (2.3.4) it is not difficult to show that

$$\begin{aligned} V(t) = {} & (1-\lambda_t)[\theta(t-1)-\theta^*(t)]P^{-1}(t-1)[\theta(t-1)-\theta^*(t)] \\ & + \lambda_t v^2(t) - \frac{\lambda_t(1-\lambda_t)\delta^2(t)}{(1-\lambda_t)+\lambda_t G(t)} \end{aligned} \tag{5.2.4}$$

where $V(t)$ was defined in (3.3.11). Using (2.3.5) now yields

$$\begin{aligned} V(t)-\sigma^2(t) = {} & (1-\lambda_t)[\theta(t-1)-\theta^*(t)]P^{-1}(t-1)[\theta(t-1)-\theta^*(t)] \\ & +\lambda_t(v^2(t)-\gamma^2)-(1-\lambda_t)\sigma^2(t-1) \end{aligned} \tag{5.2.5}$$

Since $\theta^*(t) \in E_t$ iff $V(t) \le \sigma^2(t)$, (5.2.2) then follows.

∇∇∇

Theorem (5.2) shows that by choosing γ^2 to be larger than the actual bound say γ'^2 on $v^2(t)$, it is possible to increase the tracking capability of the algorithm. The next theorem gives an upper bound on the maximum variation in the parameters for which tracking is guaranteed.

Theorem 5.3 If $\theta^*(t-1) \in E_{t-1}$, and $\lambda_t \neq 0$, then $\theta^*(t) \in E_t$ if

$$\begin{aligned} \|\Delta(t)\| \le \frac{1}{\sqrt{\lambda_{min}[P^{-1}(t-1)]}}\Bigg[& \sqrt{\sigma^2(t-1)+\frac{\lambda_t}{1-\lambda_t}\frac{\lambda_{min}[P^{-1}(t-1)]}{\lambda_{max}[P^{-1}(t-1)]}[\gamma^2 - v^2(t)]} \\ & -\sqrt{\sigma^2(t-1)}\Bigg] \end{aligned} \tag{5.2.6}$$

where

$$\Delta(t) = \theta^*(t) - \theta^*(t-1) \tag{5.2.7}$$

and λ_{min} and λ_{max} denote minimum and maximum eigenvalues respectively.

Proof: Using (5.2.7), it is straightforward to show that

$$[\theta(t-1)-\theta^*(t)]P^{-1}(t-1)[\theta(t-1)-\theta^*(t)]$$
$$= V(t-1)+\Delta^T(t)P^{-1}(t-1)\Delta(t)+2\Delta^T(t)P^{-1}(t-1)\tilde{\theta}(t-1) \tag{5.2.8}$$

where $\tilde{\theta}(t-1) = \theta(t-1) - \theta^*(t-1)$. Now substituting (5.2.8) in (5.2.5) yields

$$V(t)-\sigma^2(t) = (1-\lambda_t)[V(t-1)-\sigma^2(t-1)]+\lambda_t(\gamma'^2-\gamma^2)$$
$$+(1-\lambda_t)[\Delta^T(t)P^{-1}(t-1)\Delta(t)+2\Delta^T(t)P^{-1}(t-1)\tilde{\theta}(t-1)] \tag{5.2.9}$$

Since $\theta^*(t-1) \in E_{t-1}$, therefore $V(t-1) \le \sigma^2(t-1)$ and thus a sufficient condition for $\theta^*(t) \in E_t$ is

$$\Delta^T(t)P^{-1}(t-1)\Delta(t)+2\Delta^T(t)P^{-1}(t-1)\tilde{\theta}(t-1) \le \frac{\lambda_t}{1-\lambda_t}(\gamma^2-\gamma'^2) \tag{5.2.10}$$

i.e. $\theta^*(t) \in E_t$ if

$$\lambda_{max}[P^{-1}(t-1)]\|\Delta(t)\|^2+2\|\Delta(t)\|\,\|\tilde{\theta}(t-1)\|\lambda_{max}[P^{-1}(t-1)]$$
$$\le \frac{\lambda_t}{1-\lambda_t}(\gamma^2-\gamma'^2) \tag{5.2.11}$$

Since $V(t-1) \le \sigma^2(t-1)$, therefore

$$\|\tilde{\theta}(t-1)\|^2 \le \frac{\sigma^2(t-1)}{\lambda_{min}[P^{-1}(t-1)]} \tag{5.2.12}$$

Substituting (5.2.12) in (5.2.11) gives a sufficient condition for $\theta^*(t) \in E_t$ as

$$\lambda_{max}[P^{-1}(t-1)]\|\Delta(t)\|^2+2\|\Delta(t)\|\sqrt{\sigma^2(t-1)}\,\frac{\lambda_{max}[P^{-1}(t-1)]}{\sqrt{\lambda_{min}[P^{-1}(t-1)]}}$$
$$\le \frac{\lambda_t}{1-\lambda_t}(\gamma^2-\gamma'^2) \tag{5.2.13}$$

Solving this quadratic inequality then yields (5.2.6).

∇∇∇

The noise term $v^2(t)$ in 5.2.6 can be replaced by γ^2 to yield a bound which can be calculated at the time instant t. If $\lambda_t = 0$, then the difference between γ^2 and γ'^2 cannot be exploited to increase the tracking capability of the algorithm. In fact in this case, $\theta^*(t) \in E_t$ iff $\theta^*(t) \in E_{t-1}$. Thus if $\theta^*(t)$ jumps out of E_{t-1}, and no updates are performed at future time instants $t+i$, then $\theta^*(t+i) \notin E_{t+i} = E_{t-1}$, and the parameter may never be tracked. However, it can be argued that an update will be performed in a finite interval of time .

This is shown heuristically, by examining the expression for the magnitude of the prediction error

$$|\delta(t)| = | [\theta(t-1) - \theta^*(t)]^T\Phi(t) + v(t) |$$

If no updates are performed for a large interval of time say from time instant t, to time instant $t + N_1$ then

$$| [\theta(t-1) - \theta^*(t+i)]^T\Phi(t+i) + v(t+i) | \le [\gamma^2 - \sigma^2(t-1)]^{1/2} \quad \forall\ i = 0,1,\ldots,N_1.$$

If the input and noise sequences are sufficiently rich, then the regressor vector $\Phi(t)$ will span the parameter space in all directions and so $[\theta(t-1) - \theta^*(t+i)]^T\Phi(t+i)$ cannot be arbitrarily small for $i \in [0, N_1]$. If $|v(t+i)|$ approaches its upper bound γ, for some i in the same interval, and if $\{v(t)\}$ is sufficiently uncorrelated with the input $\{u(t)\}$, then the above inequality will be violated and an update will be performed.

If the parameter variation is such that (5.2.2) is violated then $\theta^*(t) \notin E_t$. The next theorem shows that if $\theta^*(t)$ remains fixed after its jump out of E_t, and if the jump is not large enough to cause the subsequent ellipsoids E_{t+i} , for $i \ge 0$, to vanish, then the DHOBE algorithm guarantees that the true parameter will be tracked (enclosed) in finite time.

Theorem 5.4 Assume that the parameter variation at time instant t, causes $\theta^*(t) \notin E_t$. Assume further that

(1) After this variation , the parameter remains constant (i.e. the jump parameter case).

(2) $\sigma^2(t+i) > 0$, for $i \ge 0$.

(3) The algorithm does not stop updating.

(4) A lower bound ρ is imposed on λ_t at the updating instants.

Then there exists a $N_1 > 0$, which depends on the amount of parameter variation and the actual and user set noise bounds such that $\theta^*(t) \in E_{t+N_1}$.

Proof: Since $\theta^*(t) \notin E_t$, define

$$\eta = [\theta(t) - \theta^*(t)]P^{-1}(t)[\theta(t) - \theta^*(t)] - \sigma^2(t) > 0 \qquad (5.2.14)$$

Assumption (1) will imply that $\Delta(t+N_1) = \Delta(t+1) = 0$ for arbitrary positive N_1. Substituting in (5.2.9), and iterating yields

$$V(t+N_1) - \sigma^2(t+N_1) = \eta \prod_{i=t+1}^{t+N_1}(1-\lambda_i) + \sum_{i=t+1}^{t+N_1} q_{i,t+N_1}[v^2(t) - \gamma^2] \qquad (5.2.15)$$

where $q_{i,j}$ is defined in Appendix 4A. Assumption (3) will ensure that some of the λ_{t+i}, $i \geq 0$, will be non-zero. Since the second term on the right hand side of (5.2.15) is negative, and since the first term is non-increasing, the difference $V(t+N_1) - \sigma^2(t+N_1)$ will tend to zero as N_1 increases. Thus there exists a N_1 such that

$$V(t+N_1) - \sigma^2(t+N_1) \leq 0 \qquad (5.2.16)$$

An estimate of the time N_1 required for the ellipsoids to regain the parameter vector can be obtained by the following analysis.

Assume that there are K updates in the interval $[t+1, t+N_1]$. From (5.2.15), it is clear that the inequality (5.2.16) will be satisfied for all K which satisfy

$$\eta \prod_{j=1}^{K}(1-\lambda_{t_j}) \leq \sum_{j=1}^{K} q_{t_j,t_K}[\gamma^2 - \gamma'^2] \qquad (5.2.17)$$

where $\{t_j\}$ is the sequence of updating instants. Now let $R(K) = \sum_{j=1}^{K} q_{t_j,t_K}$, where the sum is taken over the updating instants. It is easy to see that

$$R(K) = (1-\lambda_{t_K})R(K-1) + \lambda_{t_K} = R(K-1) + \lambda_{t_K}[1-R(K-1)]$$

In Appendix 4A, it is shown that $R(t) < 1$ for all $t \geq 0$, and by Assumption (4), $\lambda_{t_K} \geq \rho$, hence

$$R(K) \geq R(K-1) + \rho[1-R(K-1)]$$

and so

$$R(K) \geq (1-\rho)R(K-1) + \rho$$

Thus

$$R(K) \geq \rho + \rho(1-\rho) + \ldots + \rho(1-\rho)^{K-1} = 1-(1-\rho)^K$$

Thus from (5.2.17), K has to satisfy

$$\eta(1-\rho)^K \leq [1-(1-\rho)^K](\gamma^2 - \gamma^2)$$

Hence $\theta^*(t) \in E_{t+N_1}$ if the number of updates K, in the interval $[t, t+N_1]$ satisfies

$$K \geq \frac{\log\left(\frac{\eta}{\gamma^2 - \gamma'^2} + 1\right)}{-\log(1-\rho)} \tag{5.2.18}$$

$\nabla\nabla\nabla$

5.3 A Rescue Procedure

In many cases when the parameter jump is large, or if the ellipsoid has shrunk to a very small size, the intersection of E_{t-1} and S_t can be void. In that case, $\sigma^2(t)$ can become negative, thus indicating that a bounding ellipsoid cannot be constructed. To circumvent this failure of the algorithm, a rescue procedure is proposed. If at any time instant t, $\sigma^2(t)$ becomes negative, then $\sigma^2(t-1)$ is increased by an appropriate amount, thereby increasing the size of E_{t-1}. Then the intersection of the larger E_{t-1} and S_t will no longer be void, and thus an ellipsoid E_t will be constructed. Alternatively, [illegible] ld be increased, to permit a non-null intersection. However, the former procedure is preferable because it causes $\theta(t)$ to migrate towards $\theta^*(t)$, thereby reducing the parameter estimation error.

There are essentially two different cases which require calculation of the amount of increase.

<u>Case 1</u>. $1+\beta(t)[G(t)-1] > 0$ and $\nu_t < \alpha$.

The quantities $\beta(t)$, $G(t)$, ν_t, and α have been defined in Section 2.3. In this case, $\frac{d\sigma^2(t)}{d\lambda_t}\Big|_{\lambda_t = \nu_t} = 0$, and so as in Lemma 3.1,

$$\sigma^2(t) + \varepsilon^2(t) = \gamma^2 \tag{5.3.1}$$

Thus $\sigma^2(t)$ is negative iff $|\varepsilon(t)| > \gamma$. Using (3.3.6) this implies that $\sigma^2(t)$ is negative iff

$$|\delta(t)| > \frac{1-\lambda_t+\lambda_t G(t)}{1-\lambda_t}\gamma \tag{5.3.2}$$

On substituting for λ_t from (2.3.13), (5.3.2) can be expressed as

$$|\delta(t)| > \frac{G(t)-1}{\sqrt{G(t)(1+\beta(t)[G(t)-1])}-1}\gamma \qquad \text{if } G(t) \neq 1$$

(5.3.3)

and

$$|\delta(t)| > \frac{2\gamma}{1+\beta(t)} \qquad \text{if } G(t) = 1$$

Using the definition of $\beta(t)$ from (2.3.17) in (5.3.3) and manipulating terms yields a necessary and sufficient condition for $\sigma^2(t)$ to be negative, in terms of $\sigma^2(t-1)$

$$\sigma^2(t-1) < \frac{1}{G(t)-1}\left[\delta^2(t)+\gamma^2[G(t)-1]-\frac{\gamma[G(t)-1]+|\delta(t)|}{\sqrt{G(t)}}\right] = K_1 \qquad \text{if } G(t) \neq 1$$

and (5.3.4)

$$\sigma^2(t-1) < \delta^2(t) + \gamma^2 - 2\gamma|\delta(t)| = K_1 \qquad \text{if } G(t) = 1$$

Thus the rescue procedure would replace $\sigma^2(t-1)$ by $K_1 + \zeta$, where ζ is a positive constant, to ensure that $\sigma^2(t)$ is positive. Using $\zeta = 1$, has yielded satisfactory results in simulations.

Case 2. $\lambda_t = \alpha$

In this case, from (2.3.5), $\sigma^2(t)$ is negative iff

$$\delta^2(t) \geq [1-\alpha+\alpha G(t)]\left[\frac{\sigma^2(t-1)}{\alpha}+\frac{\gamma^2}{1-\alpha}\right] \tag{5.3.5}$$

Thus $\sigma^2(t)$ is negative iff

$$\sigma^2(t-1) < \alpha\left[\frac{\delta^2(t)}{1-\alpha+\alpha G(t)}-\frac{\gamma^2}{1-\alpha}\right] = K_2 \tag{5.3.6}$$

In this case $\sigma^2(t-1)$ would be replaced by $K_2 + \zeta$.

5.4 Simulation Results

The tracking properties of the DHOBE algorithm are studied for an ARX(1,1) model

$$y(t) = ay(t-1) + bu(t) + v(t) \tag{5.4.1}$$

The nominal values are $a = -0.5$ and $b = 1.0$. The noise $\{v(t)\}$ and the input $\{u(t)\}$ is generated by a pseudo-random number generator with a uniform distribution in $[-1,1]$. For the DHOBE algorithm, $\alpha = 0.2$, $\gamma^2 = 1.0$, and $\sigma^2(0) = 100$. The parameters are varied as follows

Case 1. Slow variation in the parameter vector from $t = 1$.

The parameters a and b are varied by 1% for every 10 samples, starting from the first sample, and the output data $\{y(t)\}$ is generated for t =1,2,..1000. The final parameter estimation error is 7.0×10^{-3}, the final volume is 3.5×10^{-2} and the final sum of semi-axes is 0.52. All the bounding ellipsoids contained the true parameter. The parameter estimates are plotted against the true parameters in Figure 5.1. From the figure it is clear that the DHOBE algorithm can track slow time variations in the parameters.

Case 2. Slow variation in the parameter vector from $t = 500$.

The parameters a and b are varied by 1% for every 10 samples, starting from the 500th sample. The final parameter estimation error is 3.0×10^{-3} , the the final volume is 5.0×10^{-2} and the final sum of semi-axes is 0.54. All the bounding ellipsoids contained the true parameter. The parameter estimates are plotted against the true parameters in Figure 5.2. The figure shows that the algorithm can track slow time variations in the parameters even after it has "converged".

Case 3. Jump in the MA parameter at $t = 500$.

The parameter b is changed by 100% at the 500th sample, and a is kept constant at its nominal value. The true parameter vector is out of the bounding ellipsoids from $t = 500$.

to $t = 530$, after which it is regained by the bounding ellipsoids. The final parameter estimation error is 1.3×10^{-4}, the the final volume is 4.0×10^{-3} and the final sum of semi-axes is 0.14. The jump thus appears to have resulted in bounding ellipsoids with smaller sizes. The parameter estimates are plotted against the true parameters in Fig. 5.3. Fig. 5.4 shows the parameter estimates obtained for this case by applying the RLS algorithm with a forgetting factor $\lambda(t) = 0.9$. Fig. 5.5, shows the estimates when the variable forgetting factor proposed by Fortescue and Kershenbaum [Goodwin, 1983] is incorporated into the RLS algorithm . This variable forgetting factor $\lambda(t)$, is a function of the prediction error and is given by

$$\lambda(t) = 1 - \alpha' \frac{\delta^2(t)}{1 + G(t)}$$

A value of $\alpha' = 0.01$ was used because it yielded steady state tracking error of about the same magnitude as the DHOBE algorithm. From these figures, it is evident that the DHOBE algorithm can track jumps in the parameters at least as well as the exponentially weighted RLS algorithm.

The effect of varying γ^2 was studied. A value of $\gamma^2 = 2$ was taken. In this case, the true parameter did not jump out of the bounding ellipsoid at $t = 500$. The parameter estimates are identical to those in Fig. 5.3. But the ellipsoids are larger, as expected, with the final volume = 3.4 and sum of semi axes = 4.08

For a different run, i.e. with a different input and noise sequence, the jump at t = 500, caused $\sigma^2(t)$ to become negative. The rescue procedure was then used with remarkable results. The true parameter was captured at $t = 501$. The final parameter estimation error = 2.4×10^{-4}, the final volume = 5.8×10^{-2}, and the final sum of semi-axes= 0.65. Fig. 5.6 shows that the parameters are tracked extremely rapidly in this case.

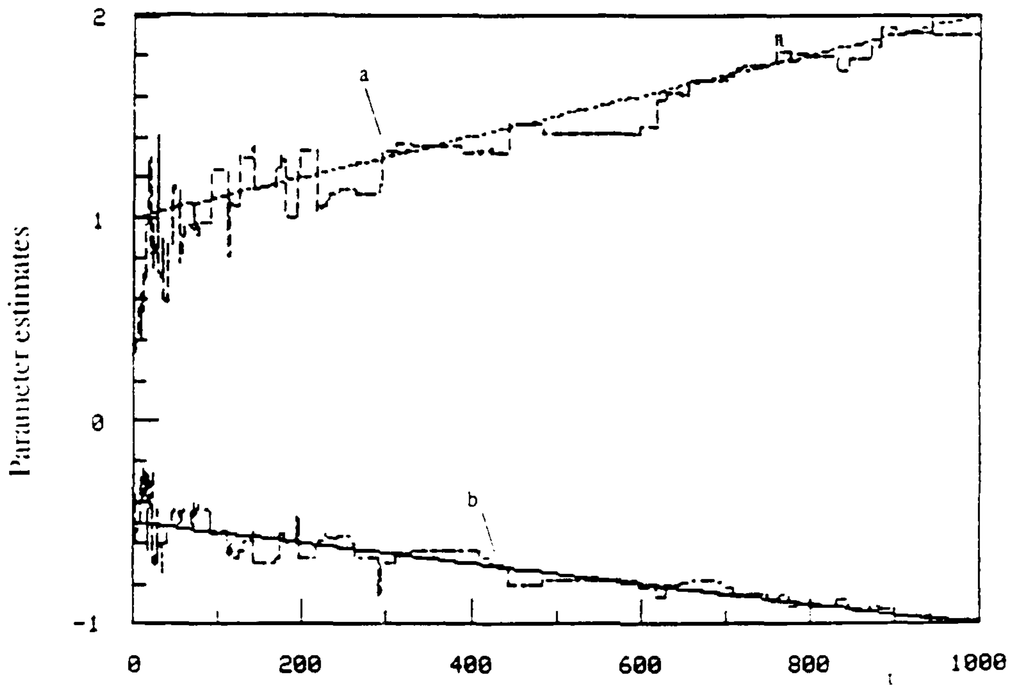

Figure 5.1 DHOBE parameter estimates for the case of slow variation in the true parameter from $t = 1$

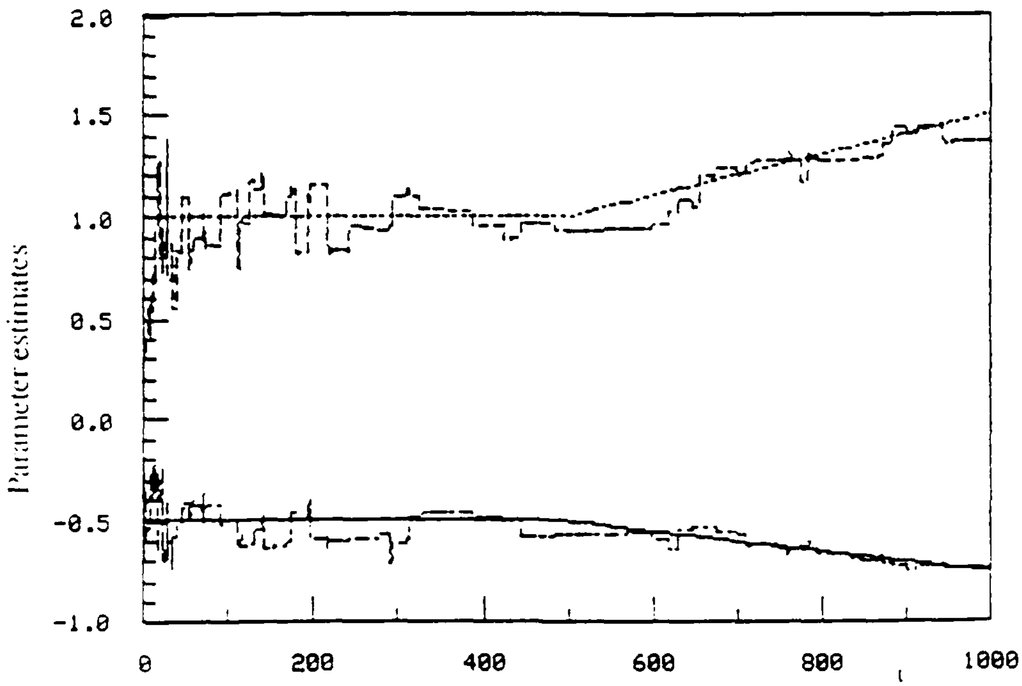

Figure 5.2 DHOBE parameter estimates for the case of slow variation in the true parameter from $t = 500$

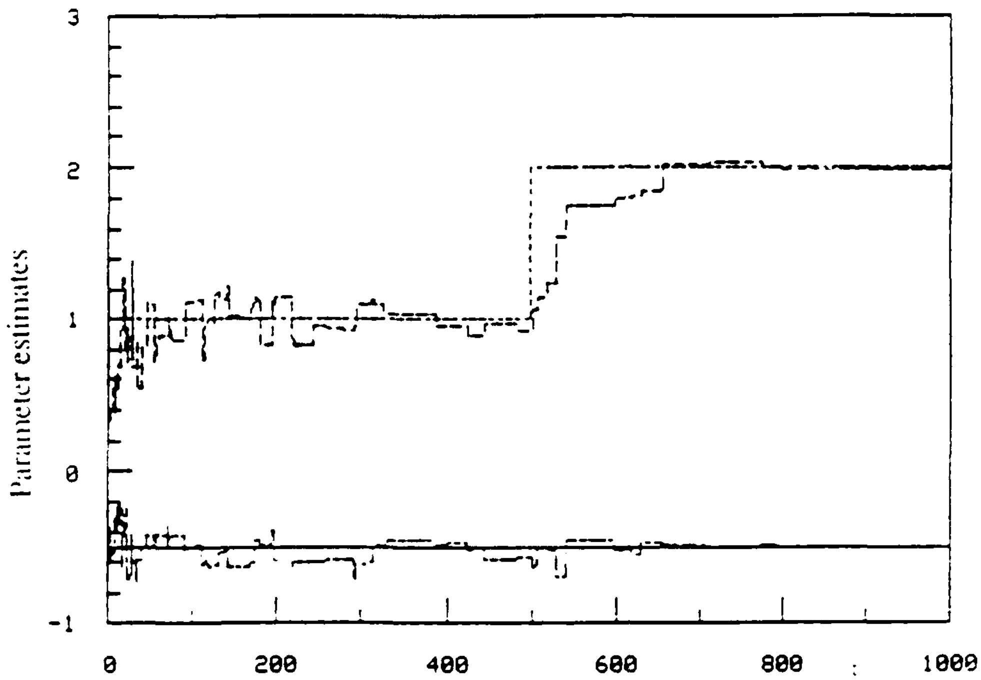

Figure 5.3 DHOBE parameter estimates for the case of a jump in the MA parameter at t = 500

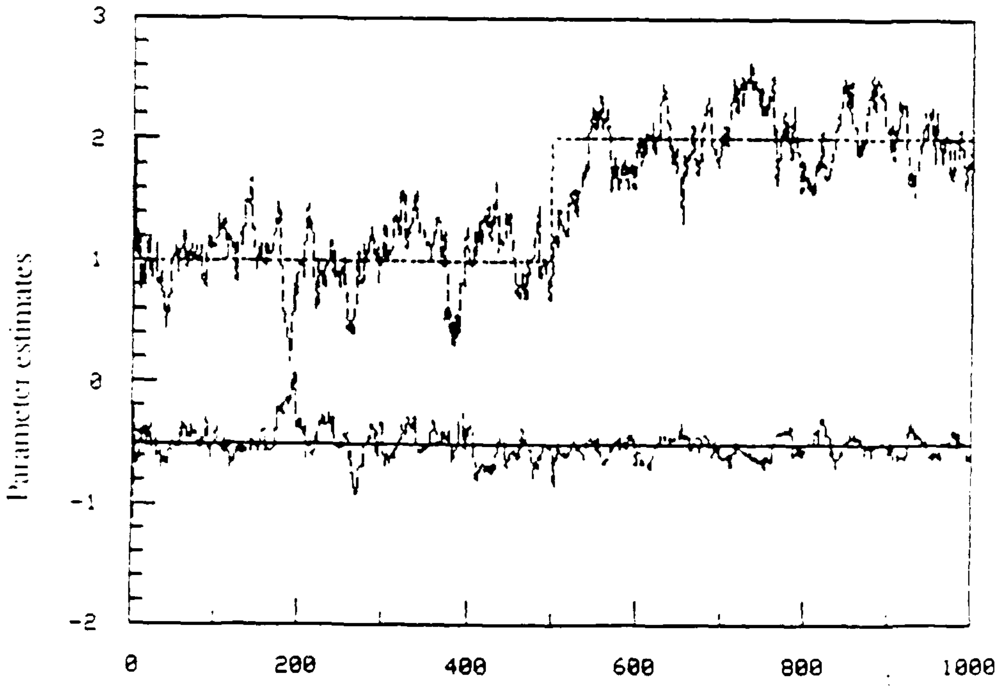

Figure 5.4 RLS (with $\lambda(t)=0.9$) parameter estimates for the jump parameter case

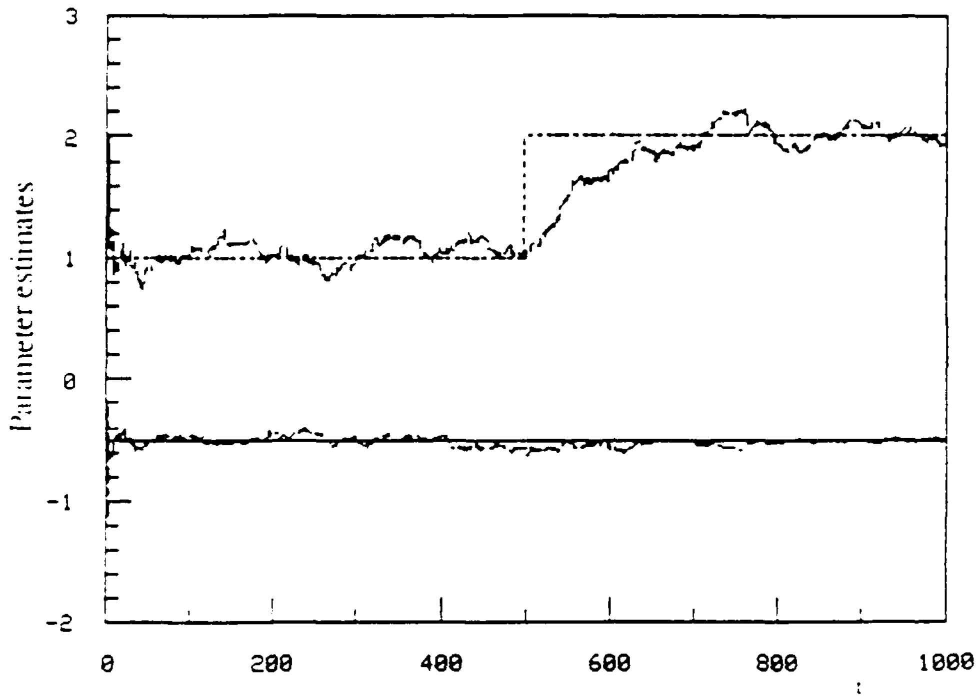

Figure 5.5 RLS (with variable forgetting factor) parameter estimates for the jump parameter case

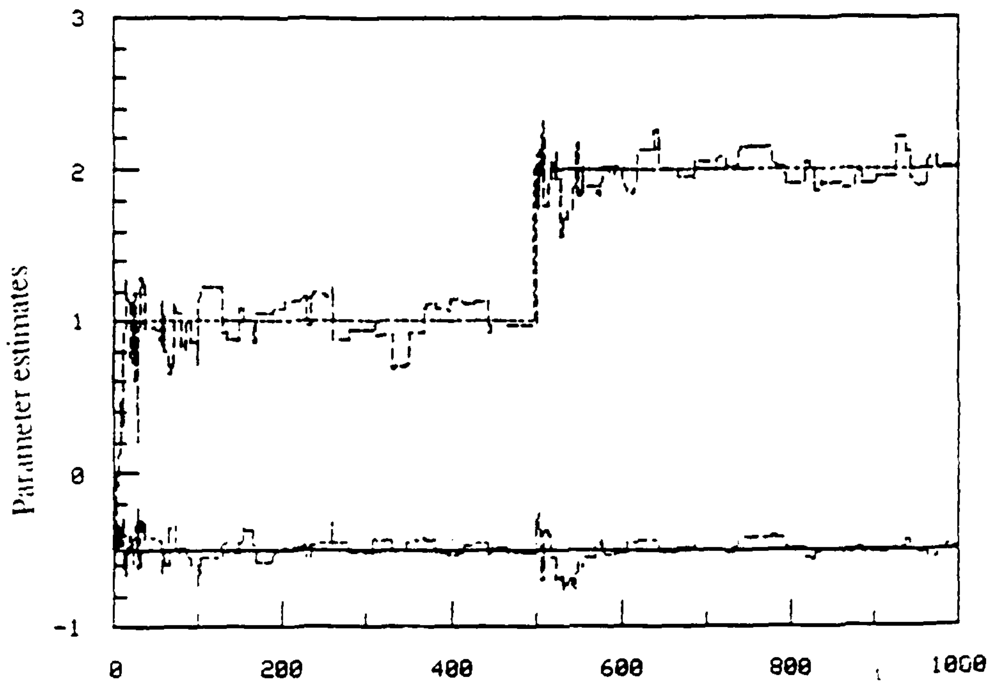

Figure 5.6 DHOBE parameter estimates when the rescue procedure is activated in the jump parameter case

CHAPTER VI

CONCLUSION

This report has focussed in on the bounding ellipsoid approach to membership-set parameter estimation. It has been shown that the OBE algorithms often yield lower estimation error in comparison to least-squares algorithms, when the unknown-but-bounded noise does not satisfy the usual stationary and whiteness assumptions. The OBE algorithms are thus viable alternatives to conventional parameter estimation algorithms in many real life applications.

Previous work in the area of membership-set parameter estimation has concentrated on parameter estimation of models with known inputs and outputs. In this report, one of the OBE algorithms- the DHOBE algorithm- has been extended to perform parameter estimation of linear models with unknown-but-bounded inputs. The extended algorithm possesses all the advantageous features of the OBE algorithms such as a discerning update strategy, time varying parameter tracking capability and robustness to numerical effects. The transient performance of the algorithm has been observed to be superior to that of the ELS algorithm in simulations. This is particularly advantageous when the number of data points is small. Analysis of the extended algorithm has shown that the algorithm yields 100% confidence intervals for the parameters at every sampling instant. The analysis of the extended algorithm requires less restrictive assumptions than the analyses of the extended least-squares or recursive maximum likelihood algorithms.

Analysis of the finite precision effects in the DHOBE algorithm has shown that the algorithm is stable with respect to errors due to finite wordlength computations and storage. A detailed analysis of this nature has never been performed for any of the other existing MSPE algorithms. Furthermore, simulation results show that the matrix recursion involved in the DHOBE algorithm is better conditioned numerically than the corresponding recursion in the conventional recursive least-squares algorithm.

The tracking characteristics of the DHOBE algorithm have been studied in detail. Some necessary and sufficient conditions for parameter tracking have been derived. It has been shown theoretically and through simulations that the algorithm can track small variations in the parameters. A procedure has been suggested, whereby large jumps in the parameters can also be tracked.

The work performed in this report can provide a spring board for several areas of future research. The connection between the OBE algorithms and the weighted least-squares algorithm could perhaps be exploited to develop fast ($O(N)$) implementations. It may turn out that the numerical properties of these implementations are superior to those of the existing fast least-squares algorithms. Another promising research direction is to recast the OBE algorithms in an output error formulation. The algorithms could then be used to obtain unbiased estimates for ARX models with output noise, which are commonly used models in adaptive control. The use of the OBE algorithms in reduced order modeling is also an interesting application. If a bound on the maximum allowable modeling inaccuracy is specified, the OBE algorithms could be used to generate a class of reduced order models which can approximate the unknown large order system. Thus there exists a gamut of applications where the OBE algorithms can be applied and in fact prove to be viable alternatives to conventional parameter estimation algorithms.

APPENDIX 2A

Derivation of the Bias Expression (2.5.4)

The system model is an ARX(1,1) model

$$y(t) = x\, y(t-1) + b_0\, u(t) + v(t) \tag{2A.1}$$

where the measurable input u(t) is white and uncorrelated with the noise v(t), and

$$v(t) = A \sin(\omega_0 t) + (1-A)\, w(t) \tag{2A.2}$$

with w(t) being a white noise sequence.

Define

$$n(t) = A \sin(\omega_0 t) \tag{2A.3}$$

The predictor model is

$$\hat{y}(t) = a\, y(t-1) + b\, u(t) \tag{2A.4}$$

The RLS algorithm, at time instant t = N minimizes

$$\frac{1}{N}\sum_{t=1}^{N}\left[y(t) - \hat{y}(t)\right]^2 \tag{2A.5}$$

The RLS solution thus satisfies

$$\frac{1}{N}\sum_{t=1}^{N}\left[(x-a)y(t-1) + (b_0 - b)u(t) + v(t)\right]\begin{bmatrix} y(t-1) \\ u(t) \end{bmatrix} = 0 \tag{2A.6}$$

In order to obtain an expression for the asymptotic bias the following definitions are made.

Let p(t) and q(t) be two signals. Define the sample expectation

$$\overline{E}[p(t)] = \lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N} p(t) \tag{2A.7}$$

and the sample cross correlation

$$\bar{R}_{pq}(i) = \bar{E}[p(t-i)q(t)] = \lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} p(t-i)\,q(t) \tag{2A.8}$$

provided the limit exists. Then from (2A.6) the RLS solution asymptotically satisfies

$$a = x + \frac{\bar{R}_{yv}(1)}{\bar{R}_{yy}(0)} \tag{2A.9}$$

and

$$b = b_0 \tag{2A.10}$$

Thus the estimate of the moving average coefficient is unbiased.

To obtain an expression for the bias in the AR coefficient, the sample correlations in (2A.9) are evaluated. Multiplying the L.H. S. and R.H.S of (2A.1) by y(t) , taking sample expectations and exploiting the fact that u(t) is white and uncorrelated with v(t) yields

$$\bar{R}_{yy}(0) = x\,\bar{R}_{yy}(1) + b_0{}^2\,\sigma_u^2 + \bar{R}_{yv}(0)$$

But since w(t) is white

$$\bar{R}_{yv}(0) = \bar{R}_{yn}(0) + (1-A)^2\,\sigma_w^2$$

Hence

$$\bar{R}_{yy}(0) = x\,\bar{R}_{yy}(1) + b_0\,\sigma_u^2 + \bar{R}_{yn}(0) + (1-A)^2\,\sigma_w^2 \tag{2A.11}$$

It is easy to show [Ljung, 1987, pg. 28], that

$$\bar{E}[n^2(t)] = \frac{A^2}{2}, \tag{2A.12}$$

and

$$\bar{E}[n(t)n(t-\tau)] = \frac{A^2}{2}\cos\omega_0\tau \tag{2A.13}$$

Now multiplying both sides of (2A.1) by n(t) and taking sample expectations yields

$$\bar{R}_{yn}(0) = x\,\bar{R}_{yn}(1) + \bar{R}_{nn}(0) \tag{2A.14}$$

or alternatively

$$\bar{E}[y(t)n(t)] = \lim_{N\to\infty} y(0)\frac{1}{N}\sum_{t=1}^{N} x^t + \lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{t} x^i n(t)n(t-i) \tag{2A.15}$$

Since $|x| < 1$ to ensure a stable system, the first term on the right hand side of (2A.15) vanishes. Using the fact that the series $\sum_{i=1}^{t} x^i n(t)n(t-i)$ converges, it is not difficult to show that

$$\overline{E}[y(t)n(t)] = \lim_{N\to\infty} \sum_{i=1}^{N} x^i \frac{1}{N} \sum_{t=1}^{N} n(t)n(t-i) \tag{2A.16}$$

which using (2A.13) yields

$$\overline{E}[y(t)n(t)] = \frac{A^2}{2} \lim_{N\to\infty} \sum_{i=1}^{N} x^i \cos\omega_0 i \tag{2A.17}$$

By expressing x as e^{α}, the infinite sum in (2A.17) can be evaluated to yield

$$\overline{E}[y(t)n(t)] = \frac{A^2}{2} \frac{1 - x\cos\omega_0}{1 - 2x\cos\omega_0 + x^2} \tag{2A.18}$$

And so using (2A.12) and (2A.14)

$$\overline{E}[y(t-1)n(t)] = \overline{R}_{yn}(1) = \frac{A^2}{2} \frac{\cos\omega_0 - x}{1 - 2x\cos\omega_0 + x^2} \tag{2A.19}$$

Since w(t) is white, therefore, $\overline{R}_{yy}(1) = \overline{R}_{yn}(1)$.

Now multiplying (2A.1) by y(t-1), and taking sample expectations as in (2A.11) yields

$$\overline{R}_{yy}(1) = x\,\overline{R}_{yy}(0) + \overline{R}_{yn}(1) \tag{2A.20}$$

Using (2A.11) and (2A.20) to solve for $\overline{R}_{yy}(0)$ then yields

$$\overline{R}_{yy}(0) = \frac{1}{1 - x^2}\left[\frac{A^2[1 - x\cos\omega_0]}{1 - 2x\cos\omega_0 + x^2} + b_0^2\sigma_u^2 + (1 - A)^2\sigma_w^2 - \frac{A^2}{2}\right] \tag{2A.21}$$

Substituting the expressions for $\overline{R}_{yy}(0)$ and $\overline{R}_{yy}(1) = \overline{R}_{yn}(1)$ in (2A.9) finally yields

$$a = x + \frac{\frac{A^2}{2}[\cos\omega_0 - x]}{\frac{A^2}{2} + \left(b_0^2\sigma_u^2 + (1 - A)^2\sigma_w^2\right)\left[\frac{1 - 2x\cos\omega_0 + x^2}{1 - x^2}\right]}$$

APPENDIX 3A

Proof of the updating gain formula (3.2.8) and (3.2.9):

Since the optimum forgetting factor λ_t^* minimizes $\sigma^2(t)$, therefore

$$\sigma^2(t, \lambda_t^*) \leq \sigma^2(t, 0) = \sigma^2(t-1) \tag{3A.1}$$

and

$$\frac{d\sigma^2(t)}{d\lambda_t} = \gamma^2 - \sigma^2(t-1) - \delta^2(t)\,\frac{(1-\lambda_t)^2 - \lambda_t^2\, G(t)}{(1-\lambda_t + \lambda_t G(t))^2} \tag{3A.2}$$

and

$$\frac{d^2\sigma^2(t)}{d\lambda_t^2} = \frac{2\,\delta^2(t)\, G(t)}{(1-\lambda_t + \lambda_t G(t))^3} \tag{3A.3}$$

Thus $d^2\,\sigma^2(t) / d\,\lambda_t^2 > 0$, unless $\delta^2(t) = 0$ or $G(t) = 0$. Since $P(t-1)$ is positive definite, $G(t) = 0$ iff $\Phi(t) = 0$. The algorithm can be modified to detect the occurrence of a null $\Phi(t)$ and set it to a small non-zero value, prior to the calculation of $G(t)$. Thus it can be assumed that $G(t) \neq 0$ for all t. If $\delta^2(t) = 0$, then, since $\sigma^2(0) < \gamma^2$ by (3.2.7), and since $\sigma^2(t)$ is non-increasing, therefore $\sigma^2(t-1)+\delta^2(t) < \gamma^2$ and by (3A.2), $d\sigma^2(t)/d\lambda_t$ is positive, and hence $\sigma^2(t)$ is minimized if $\lambda_t^* = 0$. Now for the sequel, the second derivative of $\sigma^2(t)$ can be assumed to be positive and hence the unique minimum occurs at $d\,\sigma^2(t) / d\lambda_t = 0$. From (3A.2), if $G(t) = 1$, $\sigma^2(t)$ is minimized if

$$\lambda_t^* = (1 - \beta(t))/2 \tag{3A.4}$$

Otherwise if $G(t) \neq 1$, $\sigma^2(t)$ is minimized if

$$\lambda_t^* = \frac{1}{1 - G(t)}\left[1 - \sqrt{\frac{G(t)}{1 + \beta(t)(G(t) - 1)}}\ \right] \tag{3A.5}$$

Moreover, in (3A.4) and (3A.5)

$$\lambda_t^* > 0 \Leftrightarrow \beta(t) < 1 \Leftrightarrow \sigma^2(t-1) + \delta^2(t) > \gamma^2 \qquad (3A.6)$$

It is easy to show that $1+\beta(t)\,(G(t)-1)$ is always positive. Since $\sigma^2(0) < \gamma^2$ and $\sigma^2(t)$ is non-increasing, therefore $\beta(t) > 0$. From (3A.6), $\beta(t) < 1$, hence $1- 1/\beta(t) < 0$. Then

$$1+\beta(t)\,(G(t)-1) \leq 0 \Rightarrow G(t) \leq 1- 1/\beta(t) \Rightarrow G(t) < 0$$

which is a contradiction. Thus (3A.5) would always yield real λ_t^*. It is now shown that (3A.4) and (3A.5) yield values of λ_t^* which are upper bounded by unity.

If $G(t)=1$, then since $\beta(t) > 0$, (3A.4) yields $\lambda_t^* < 1$.

If $G(t) < 1$, then $\lambda_t^* \geq 1 \Leftrightarrow 1-[\,G(t)/\,(1+ \beta(t)(G(t)-1)\,)\,]^{1/2} \geq 1- G(t)$

$$\Leftrightarrow G(t)\,(1+ \beta(t)(G(t)-1)\,) \geq 1 \qquad (3A.7)$$

But $G(t) < 1$ and $\beta(t) > 0$ contradict (3A.7). Hence if $G(t) < 1$, then $\lambda_t^* < 1$. It can be shown in exactly the same way that $G(t) > 1$, would imply that $\lambda_t^* < 1$. Thus unlike the case in [Dasgupta, 1987], no upper bound has to be imposed on the forgetting factor.

APPENDIX 3B

A Time Domain Implication of the SPR Condition

A sufficient condition for convergence of the ELS algorithm is that the transfer function $H(z^{-1}) = [\frac{1}{C(z^{-1})} - \frac{1}{2}]$ be strictly positive real . This means

$$\text{Re } H(e^{j\omega}) > 0 \quad \forall\ \omega \in [-\pi,\pi] \tag{3B.1}$$

A necessary condition for (3B.1) to hold will now be derived.

Using the definition of $H(z^{-1})$, (3B.1) becomes

$$\text{Re}\{ \frac{1}{C(e^{-j\omega})} - \frac{1}{2}] > 0 \quad \forall\ \omega \in [-\pi,\pi] \tag{3B.2}$$

Now

$$\text{Re}\{ \frac{1}{C(e^{-j\omega})} - \frac{1}{2}] = \frac{1}{2}\text{Re}\ \{\frac{1-c_1e^{-j\omega} - c_2e^{-2j\omega} - \ldots - c_re^{-rj\omega}}{1+c_1e^{-j\omega} + c_2e^{-2j\omega} - \ldots + c_re^{-rj\omega}}\}$$

$$= \frac{1}{2}\text{Re}\left\{\frac{1-c_1\cos\omega - c_2\cos 2\omega - \ldots - c_r\cos r\omega + j(c_1\sin\omega + c_2\sin 2\omega + \ldots + c_r\sin r\omega)}{1+c_1\cos\omega + c_2\cos 2\omega + \ldots + c_r\cos r\omega - j(c_1\sin\omega + c_2\sin 2\omega + \ldots + c_r\sin r\omega)}\right\}$$

Rationalizing the above expression and taking the real part yields

$$\text{Re}\left\{\frac{1}{C(e^{-j\omega})} - \frac{1}{2}\right\} = \frac{1}{x(\omega)}\left[1-\left(c_1^2 + c_2^2 + \ldots + c_r^2\right) - 2\sum_{i=1}^{j-1}\sum_{j=2}^{r} c_ic_j\cos(j-i)\omega\right] \tag{3B.3}$$

$$= \frac{1}{x(\omega)}\left[1-\left(c_1^2 + c_2^2 + \ldots + c_r^2\right) - \sum_{j=1}^{r-1} K_j\cos j\omega\right] \tag{3B.4}$$

where $x(\omega)$ is positive function of the c's and ω, and

$$K_j = 2\sum_{i=1}^{r-j} c_ic_{i+j} \tag{3B.5}$$

The SPR condition (3B.2) then implies

$$1 > \left(c_1^2 + c_2^2 + \ldots + c_r^2\right) + \sum_{j=1}^{r-1} K_j\cos j\omega \quad \forall\omega \in [-\pi,\pi]$$

Now define

$$A_{r-1}(\omega) = \sum_{j=1}^{r-1} K_j \cos j\omega \quad \forall \omega \in [-\pi, \pi]$$

Then

$$\int_{-\pi}^{\pi} A_{r-1}(\omega)\, d\omega = \sum_{j=1}^{r-1} K_j \int_{-\pi}^{\pi} \cos j\omega\, d\omega = 0$$

Hence $A_{r-1}(\omega)$ cannot have the same sign for all $\omega \in [-\pi,\pi]$. So for some ω, $A_{r-1}(\omega)$ will be non-negative, and thus

$$c_1^2 + c_2^2 + .. + c_r^2 < 1$$

Thus the SPR condition implies that the l_2 norm of the impulse response of the filter $C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + .. c_r z^{-r}$, is upper bounded by 2, or in other words, the coefficient vector $[c_1, c_2, .. c_r]^T$ lies in the unit sphere centered at the origin of r-dimensional Euclidean space.

Next, it is shown that the condition (3.3.10b) implies that the SPR condition (3B.2) will hold. It can be seen from (3B.3) that

$$\mathrm{Re}\left\{\frac{1}{C(e^{-j\omega})} - \frac{1}{2}\right\} \geq \frac{1}{x(\omega)}\left[1 - \left(c_1^2 + c_2^2 + .. + c_r^2\right) - 2\sum_{i=1}^{j-1}\sum_{j=2}^{r} |c_i|\, |c_j|\right]$$

Thus (3B.2) will hold if

$$1 - \left(c_1^2 + c_2^2 + .. + c_r^2\right) - 2\sum_{i=1}^{j-1}\sum_{j=2}^{r} |c_i|\, |c_j| \geq 0$$

i.e. if

$$1 - \left(|c_1| + |c_2| + .. + |c_r|\right)^2 \geq 0$$

Thus $\sum_{i=1}^{r} |c_i| < 1$ implies that the SPR condition (3B.2) is true.

APPENDIX 4A

Summability of the Weighting Factors of the DHOBE Algorithm

Define

$$q_{it} = \begin{cases} \lambda'_i \prod_{j=i+1}^{t} (1 - \lambda' j) & \text{if } i < t \\ \lambda'_t & \text{if } i = t \end{cases} \tag{4A.1}$$

Then the term in square brackets in (4.4.32) can be defined as

$$R(t) = \sum_{j=t_0+1}^{t} q_{jt} \tag{4A.2}$$

And

$$R(t) = (1 - \lambda'_t)R(t-1) + \lambda'_t \tag{4A.3}$$

Now

$$R(t_0+1) = \lambda'_{t_0+1} < 1$$

Assume

$$R(t-1) < 1$$

Then by (4A.3)

$$R(t) < (1 - \lambda'_t).1 + \lambda'_t$$

i.e.

$$R(t) < 1$$

REFERENCES

[**Anderson, 1983**] B. D. O. Anderson and R. M. Johnstone, " Adaptive systems and time varying plants," *Int. J. Control*, vol. 37, No. 2, pp. 367-377, 1983.

[**Belforte, 1985**] G. Belforte and B. Bona, " An improved parameter identification algorithm for signals with unknown-but-bounded errors," *Proc. 7th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, pp. 1507-1512, 1985.

[**Benveniste, 1982**] A. Benveniste and G. Ruget, " A measure of the tracking capability of recursive stochastic algorithms with constant gains," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 639-649, 1982.

[**Bierman, 1977**] G. J. Bierman, *Factorization Methods for Discrete Estimation*, Academic Press, New York 1977.

[**Cadzow, 1982**] J. A. Cadzow, " Spectral estimation : an overdetermined rational model equation approach," *Proc. IEEE* , vol. 70, No. 9, pp. 907-939, Sept 1982.

[**Cioffi, 1984**] J. M. Cioffi and T. Kailath, " Fast RLS transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, No. 2, pp. 304- 337, April 1984.

[**Cioffi, 1987**] J. M. Cioffi, "Limited-precision effects in adaptive filtering,"*IEEE Trans. Circuits and Systems*, vol. CAS-34, No. 7, pp. 821-833, July 1987.

[**Coxeter, 1973**] H. S. M. Coxeter, *Regular Polytopes*, Dover Publications, New York, 1973.

[**Dasgupta, 1987**] S. Dasgupta and Y.F. Huang, " Asymptotically convergent modified recursive least-squares..," *IEEE Trans. Information Theory* , vol. IT-33, No.3, pp. 383-392, May 1987.

[**Dasgupta, 1988**] S. Dasgupta, J. S. Garnett and C. R. Johnson, " Convergence of an adaptive filter with signed error filtering," *Proc. 1988 Amer. Contr. Conf.*, Atlanta, vol.1, pp. 400-405.

[**Deller, 1989**] J. R. Deller, " 'Systolic array ' implementations of the optimal bounding ellipsoid algorithm," *IEEE Trans. Acoust.Speech and Signal Process.*, to be published.

[**Eleftheriou, 1986**] E. Eleftheriou and D. D. Falconer, " Tracking properties and steady-state performance of RLS adaptive filter algorithms," *IEEE Trans. Acoust.Speech and Signal Process.*, vol. ASSP-34, No. 5, pp. 1097-1109, Oct. 1986.

[**Fogel, 1979**] E. Fogel, " System identification via membership set constraints with energy constrained noise, *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 752-758, 1979.

[**Fogel, 1982**] E. Fogel and Y.F. Huang, " On the value of information in system identification - bounded noise case," *Automatica* , vol. 18, No. 2, pp. 229-238, March 1982.

[**Fortescue, 1981**] T. R. Fortescue *et al.* " Implementation of self tuning regulators with variable forgetting factors," *Automatica*, vol. 17, pp. 831-835, 1981.

[**Friedlander, 1982**] B. Friedlander, "System identification techniques for adaptive signal processing, "*IEEE Trans. Acoust. Speech and Signal Process.* , vol. ASSP-30, No. 2, pp. 240-246, April 1982.

[**Friedlander, 1984**] B. Friedlander and B. Porat, "A spectral matching technique for ARMA parameter estimation, " *IEEE Trans. Acoust. Speech and Signal Process.* , vol. ASSP-32, No. 2, pp. 338-343, April 1984.

[**Goodwin, 1984**] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[**Goodwin, 1986**] G. C. Goodwin *et al*, " Sinusoidal disturbance rejection with application to helicopter flight data estimation," *IEEE Trans. Acoust. Speech and Signal Process.* , vol. ASSP-34, No. 3, pp. 479-484, 1986.

[**Gunnarsson, 1989**] S. Gunnarsson and L. Ljung, " Frequency domain tracking characteristics of adaptive algorithms," *IEEE Trans. Acoust.Speech and Signal Process.*, vol. ASSP-37, No. 7, pp.1072-1089, July 1989.

[**Huang, 1980**] Y. F. Huang, " System Identification using membership-set description of system uncertainties and a reduced-order optimal state estimator," M.S. Thesis, Dept. of Electrical Engg., Univ. of Notre Dame, IN, Jan. 1980.

[**Huang, 1986**] Y. F. Huang, " A recursive estimation algorithm using selective updating for spectral analysis and adaptive signal processing,"*IEEE Trans. Acoust. Speech and Signal Process.* , vol. ASSP-34, No. 5, pp. 1131-34 , October 1986.

[**Huang, 1987**] Y. F. Huang and R. W. Liu, " A high level parallel-pipelined network architecture for adaptive signal processing, " in *Proc. 26th IEEE Conf. on Decision and Control*, vol.1, pp. 662-666, Dec. 1987.

[**Kaveh, 1979**] M. Kaveh, " High resolution spectral estimation for noisy signals," *IEEE Trans. Acoust. Speech and Signal Process.* , vol. ASSP-27, No. 3, pp. 286-287, June 1979.

[**Ljung, 1983**] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, Mass. 1983.

[**Ljung, 1985**] S. Ljung and L. Ljung, "Error propagation properties of recursive least-squares adaptation algorithms," *Automatica*, vol. 21, pp. 157-167, 1985.

[**Ljung, 1987**] L. Ljung , *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[**Mayne, 1982**] D. Q. Mayne *et al*, " Linear identification of ARMA processes," *Automatica*, vol. 18, No. 4, pp. 461-466, July 1982.

[**Messerschmitt, 1984**] D. G. Messerschmitt, " Echo cancellation in speech and data transmission," *IEEE J. Select Areas Commun.*, vol. SAC-2, pp. 283-297, March 1984.

[**Milanese, 1982**] M. Milanese and G. Belaforte, " Estimation theory and uncertainty intervals," *IEEE Trans. Automatic Control* , vol. AC-27, No. 2, pp. 408-414, April 1982.

[**Mo, 1988**] S. H. Mo and J. P. Norton, " Recursive parameter-bounding algorithms which compute polytope bounds,"*Proc. 12th IMACS World Congress*. Paris, France, July 1988, pp. 477-480.

[**Pearson, 1986**] R. Pearson, " Sequential algorithms for set-theoretic estimation," *SIAM Journal on Algebraic and Discrete Methods*," 1988.

[**Piet-Lahanier, 1988**] H. Piet-Lahanier and E. Walter, " Practical implementation of an exact and recursive algorithm for characterizing likelihood sets," *Proc. 12th IMACS World Congress*. Paris, France, July 1988, pp. 481-483.

[**Rao, 1988**] B. D. Rao and R. Peng, " Tracking characteristics of the constrained IIR adaptive notch filter," *IEEE Trans. Acoust.Speech and Signal Process.*, vol. ASSP-36, No. 9, pp. 1466-1479, Sept. 1988.

[**Schweppe, 1968**] F. C. Schweppe, " Recursive state estimation : unknown but bounded errors and system inputs, " *IEEE Trans. Automatic Control*, vol. AC-13, No.1, pp. 22-28, Feb. 1968.

[**Walter, 1987**] E. Walter and H. Piet-Lahanier," Exact and recursive description of the feasible parameter set for bounded error models," *Proc. 26th Conference on Decision and Control*, Los Angeles, pp. 1921-1922, 1987.

[**Widrow, 1976**] B. Widrow *et al.* " Stationary and non stationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, pp. 1151-1162, 1976.